

SANDIA REPORT

SAND2001-0684

Unlimited Release

Printed March 2000

Final Report and Documentation for the Optical Backplane/Interconnect for High Speed Communication LDRD

Perry J. Robertson, Helen Y. Chen, James M. Brandt, Charles T. Sullivan,
Lyndon G. Pierson, Edward L. Witzke, and Karl Gass

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia is a multiprogram laboratory operated by Sandia Corporation,
a Lockheed Martin Company, for the United States Department of
Energy under Contract DE-AC04-94AL85000.

Approved for public release; further dissemination unlimited.



Sandia National Laboratories

Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government, nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, make any warranty, express or implied, or assume any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represent that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof, or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof, or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865)576-8401
Facsimile: (865)576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: <http://www.doe.gov/bridge>

Available to the public from
U.S. Department of Commerce
National Technical Information Service
5285 Port Royal Rd
Springfield, VA 22161

Telephone: (800)553-6847
Facsimile: (703)605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: <http://www.ntis.gov/ordering.htm>



SAND2001-0684
Unlimited Release
Printed March 2001

**Final Report and Documentation for the
Optical Backplane/Interconnect for High Speed Communication LDRD**

Perry J. Robertson
RF and Opto Microsystems

Helen Y. Chen and James M. Brandt
Security and Network Research

Charles T. Sullivan
Advanced Semiconductor Technology

Lyndon G. Pierson and Edward L. Witzke
Advanced Networking Integration

Sandia National Laboratories
P. O. Box 5800
Albuquerque, New Mexico 87185-0874

Karl Gass
Chocolate Donut Labs, Albuquerque, NM 87110

Abstract

Current copper backplane technology has reached the technical limits of clock speed and width for systems requiring multiple boards. Currently, bus technology such as VME and PCI (types of buses) will face severe limitations as the bus speed approaches 100 MHz. At this speed, the physical length limit of an unterminated bus is barely three inches. Terminating the bus enables much higher clock rates but at drastically higher power cost. Sandia has developed high bandwidth parallel optical interconnects that can provide over 40 Gbps throughput between circuit boards in a system. Based on Sandia's unique VCSEL (Vertical Cavity Surface Emitting Laser) technology, these devices are compatible with CMOS (Complementary Metal Oxide Semiconductor) chips and have single channel bandwidth in excess of 20 GHz. In this project, we are researching the use of this interconnect scheme as the physical layer of a greater ATM (Asynchronous Transfer Mode) based backplane. There are several advantages to this technology including small board space, lower power and non-contact communication. This technology is also easily expandable to meet future bandwidth requirements in excess of 160 Gbps sometimes referred to as UTOPIA 6. ATM over optical backplane will enable automatic switching of wide high-speed circuits between boards in a system. In the first year we developed integrated VCSELs and receivers, identified fiber ribbon based interconnect scheme and a high level architecture. In the second year, we implemented the physical layer in the form of a PCI computer peripheral card. A description of future work including super computer networking deployment and protocol processing is included.

Acknowledgments

The authors would like to express their gratitude to Larry Pucket for his layout of the POI board. This board required careful attention to high-speed design, knowledge of board buildups requiring exotic dielectric materials, and aggressive high density ball grid array routing, all of which were handled with speed and efficiency. The authors would also like to acknowledge the engineers at ZTEC, Inc., Albuquerque, New Mexico for their help in the design and layout of the POI board.

Contents

1. INTRODUCTION	1
1.1 PROJECT OVERVIEW	1
1.2 DESCRIPTION OF WORK YEAR 1	3
1.3 DESCRIPTION OF WORK YEAR 2	3
1.4 ACCOMPLISHMENTS	4
2. OPTICAL BUS PHYSICAL LAYER.....	5
3. ATM BUS PROTOCOL	7
4. PHYSICAL LAYER PARALLEL OPTICAL DEVICES	9
5. OPTICAL BUS / INTERCONNECT SYSTEM ARCHITECTURE	11
6. PARALLEL OPTICAL INTERCONNECT PCI CIRCUIT BOARD.....	13
6.1 GENERAL	13
6.2 BOARD FEATURES	15
6.3 FIBER OPTIC INTERFACE.....	15
6.4 BOARD DESIGN AND LAYOUT.....	16
6.5 DATA PROCESSING	17
6.6 MISCELLANEOUS FEATURES.....	18
6.7 MECHANICAL & POWER	18
7. FUTURE DEVELOPMENT ACTIVITIES	19
7.1 SYSTEM DEMONSTRATION.....	19
7.2 ASSOCIATED TASKS.....	19
8. SUMMARY	21
9. REFERENCES	23
APPENDICIES	25
APPENDIX A. BILL OF MATERIALS FOR POI BOARD.....	27
APPENDIX B. POI SCHEMATIC	29
APPENDIX C. FPGA PROGRAM.....	39
C.1 TOP LEVEL BLOCK DIAGRAM.....	41
C.2 COMMON VHDL	43
C.3 TIMEBASE VHDL.....	45
C.4 LVDS_RX VHDL	49
C.5 LVDS_TX VHDL	51
C.6 LED_BAR VHDL	53
C.7 LED_7SEG VHDL	57
C.8 LED_DECIMAL VHDL	61
C.9 PCI INTERFACE.....	63
C.10 SDRAM CONTROLLER	63

(This page intentionally left blank)

Figures

FIGURE 1. INTEGRATED VCSEL AND RECEIVER DEVICES.	10
FIGURE 2. WAVELENGTH ADDRESSABLE CLUSTER INTERCONNECT USING FAST TUNABLE LASER AND FILTER	11
FIGURE 3. STRUCTURE OF A FRAME.....	12
FIGURE 4. TOKEN PROCESSOR.....	12
FIGURE 5. OUTLINE DRAWING OF THE POI BOARD, FRONT VIEW.	13
FIGURE 6. OUTLINE DRAWING OF THE POI BOARD, REAR VIEW.	14
FIGURE 7. BLOCK DIAGRAM OF THE POI BOARD.	14
FIGURE 8. PRINTED CIRCUIT BOARD CROSS SECTION.	16
FIGURE 9. HIGH SPEED LVDS PRINTED CIRCUIT LINES.....	17

(This page intentionally left blank)

1. Introduction

1.1 Project Overview

When this project was conceived over three years ago, there was very little interest in the industry for multiple, parallel optics solutions for large bandwidth network applications. The large carriers such as AT&T, Northern Telecom and Sprint were standing firm in the belief that their customers would not have bandwidth demands greater than 2.5 Gbps that could not be accommodated using serial fiber links. All interest in greater bandwidth concentrated on multi wavelength, wavelength division multiplexing, or WDM solutions. The industries view was that bandwidth requirements in excess of 2.5 Gbps would not develop for over five years.

The DOE ASCI program had just developed some metrics and derived requirements driven from the needs of remote visualization for nuclear weapons development that projected the need for over 100 Gbps in the very near term, three to five years. There was then quite a different picture of network bandwidth needs within the DOE community. (Not surprisingly, the rest of the industry has found out that they too need additional bandwidth in the mean time.) In seeking greater bandwidth, this project proposed to examine how Sandia developed technology, Vertical Cavity Surface Emitting Lasers (VCSELs) could be used to achieve greater bandwidth between computers.

High speed VCSEL technology has been in development at Sandia for the past decade [1]. Starting as single devices, it was quickly recognized that these devices could be arranged in arrays (both linear and rectangular). During the decade of the 90's Sandia made great strides in developing high efficiency devices, achieving wall plug efficiencies of over 50%. As early as 1994, Sandia had produced VCSEL arrays [2]. It was perceived that VCSELs in small linear arrays (8 to 16 devices) could be used with fiber ribbon to increase the bandwidth between two computers over that available from just a single device. VCSELs can be made as single mode devices compatible with single mode or 50 or 62.5 um core multimode plastic fiber (or graded index multimode fiber). Ribbon connector systems were being introduced that would handle 8 or 12 channels.

By the Fall of 2000, it was apparent that the industry view of parallel optics had changed. There are several vendors currently offering components supporting multiple VCSEL links. Vitesse and Conexant Both have devices which can be used to support multiple VCSELs. Altera has developed a Programmable Logic Device (PLD) that has 16 high speed transmitter and 16 receiver pairs capable of operating at over 1.25 Gbps. The 20K400 device from Altera, used on the POI board as part of this research effort, is a good example of the state-of-the-art of high speed interfaces. Even the logic family, Low Voltage Differential Signaling (LVDS), was not in common use three years ago [3][4]. This interface has been introduced into many high speed devices supporting high speed standards from international standard bodies such as the ATM Forum (ATM stands for Asynchronous Transfer Mode) and the Optical Internetworking Forum (OIF) [5]. Sandia has been active in both these Forums.

There is still a great wave of support among the Carriers (AT&T, Sprint, etc.) for high speed serial communications links utilizing 1310 nm single mode lasers and fibers. Many of the new

components in this area will operate at OC-192 (10 Gbps) rates. Components needed in this system are laser transmitter, serializer, deserializer, clock-and-data recovery and optical-electrical-optical conversion. These components were not available even two years ago. As mentioned above, the forecasts of carriers did not foresee the need for OC-192 serial components for over 5 years. Now, it appears that OC-768 (40 Gbps) serial components will be available by the second calendar quarter of 2001. Network Elements, San Mateo, California, expects to field such products. High level parts, framers and integrated packet processors will follow shortly afterward [6]. There will be much competition between those supporting lower speed parallel solutions and the higher cost, higher speed serial solutions in the next few years.

There are many examples of parallel optics components. Stratus Lightwave LLC, Chicago, Illinois, sells a MP "Ribbon Connector System with 12 fiber capacity in a miniaturized footprint patterned after SC styles". The Quantum QX™ Backplane System, sold by Stratus, provide many cables and assemblies for backplane systems [7]. In the design of the POI board, 12 channel electro-optic devices (V23814-K1306-M230 TxDC/MUX-ENC and V23815-K1306-M230 RxDC/DEMUX-DEC) from Siemens were used. Current devices are capable of operating at up to 1.25 Gbps. VCSELs and LVDS interfaces have been shown to operate at 2.5 Gbps and it can be expected that similar 12 channel 40 Gbps parallel optical solutions will be available [8]. The four channel InfiniBand interface utilizes single channel throughputs of up to 3.125 Gbps [9]. Each channel will operate independently for an aggregate throughput of more than 12 Gbps. Research into higher bandwidth VCSELs continues and it can be expected that single devices that can operate with data rates of 20 Gbps will be available in the next few years. In addition, the price of bandwidth is falling dramatically with the introduction of parallel optics. Agilent Technologies Inc., has introduced 12 channel optical transmitter and receiver modules (HFBR-712BP and HFBR-722BP) that are expected to be introduced at \$25 to \$30/gigabit of data bandwidth [10]. These modules operate at 2.5 Gbps per channel and consume less than 2.5W when all channels are operating. Agilent also expects that 10 Gbps modules will be available in the near future.

VCSELs are only now being considered for traditional long-haul or WAN (Wide Area Network) communication systems due to the lack of 1310 nm and 1550 nm VCSELs used these single mode systems. Until now, most commercially available VCSELs have been designed to operate at 850 nm. However, new advances in VCSEL fabrication have resulted in the introduction of commercially available 1310 nm and 1550 nm devices. The technical problem of producing 1310 nm devices is that the p-type material used in the structure of the device absorbs light in the 1300 nm band greater than in the 850 nm band making it difficult to get the device to lase. One solution, developed in part by Sandia National Labs, relies on a technique called a tunnel junction. The tunnel junction allows n-type material to be used on the top and bottom of the VCSEL creating a non-absorbing mirror. It is hoped that this technique will pave the way to the production of low cost higher efficiency 1310 nm and 1550 nm lasers for commercial WAN applications [11].

The work represented here is one attempt to develop hardware and software to operate a short reach (<300m) parallel optical network connection between computers in a cluster. The work should be characterized as highly research oriented and as paving the way for future development of practical high bandwidth communications. The realities of hardware

demonstration are currently a problem. The typical Intel based personal computer platform used in super computer clusters contains a PCI bus capable of little more than 1 Gbps communications with any one card plugged into the motherboard. Next generation computers little more than double this capacity. Having network cards that support over 40 Gbps communications seem overkill at this point in time. Current high speed routers have backplanes that operate at these speeds. However, these are specialized, expensive, designs not suitable for the current generation of low cost cluster super computing platforms. It can be expected that it will only be a short time until we see computer backplanes consisting of parallel optical components capable of keeping up with the 40 Gbps network interface cards. InfiniBand is one such commercial solution operating at over 12 Gbps at this time (December, 2000). The technology is easily within the reach of 40 Gbps in the near future. The single fly in the ointment is the cost of the electro-optical components. Hopefully, wide commercial acceptance of these devices will lead to volume cost reductions.

1.2 Description of Work Year 1

The work this first year has focused on the development of concepts and protocols needed to achieve 160 Gbps (1.6×10^{11} bits per second) parallel optical interconnects for super computing and network switch applications. In the first year we accomplished the milestones set out at the beginning of this project. 1) The optical ATM (Asynchronous Transfer Mode) bus architecture has been developed. At the hardware level, commercially available fiber optic connectors have been identified that can hold parallel optical ribbon fibers in groups of twelve. Multiple connectors will be used to implement the backplane. 2) A new generation of VCSELs (Vertical Cavity Surface Emitting Laser) and receivers were developed in Sandia's CSRL (Compound Semiconductor Research Laboratory). These devices are integrated as VCSEL/receiver pairs that can either be used over separate fibers or can possibly operate bi-directionally over a single fiber. 3) Prototype hardware is being designed to fit in a computer and have two optical connections. Multiple cards are plugged into a host system and data is sent over the optical bus to exercise the architectural concepts. 4) We developed two broadcast-and-select architectures implemented as star networks that can utilize passive optical couplers and/or tunable transmitters/receivers to implement a shared memory module of a super computer. A third approach avoids some disadvantages of the previous architectures such as their lack of multicast and collision avoidance by implementing a token based control system.

1.3 Description of Work Year 2

The work the second year focused on the development of a physical layer implementation of the parallel optical interconnects. This year completed the technology evaluation and completed the design of a PCI card containing a twelve channel VCSEL based interconnect module. This board contains commercially available fiber optic connectors have been identified that can hold parallel optical ribbon fibers in groups of twelve. The board contains a single programmable logic device (PLD) that interconnects the electro-optical modules and the PCI control bus. This card was designed to operate in a PC as a network interface card (NIC). The PLD is used to implement the ATM/IP protocols and perform low level interconnect protocols on the optical link. This board was fabricated. There was not time to populate the board or perform any testing. This work will remain for a future project. A Sandia developed bit error rate core,

intellectual property (IP), has been implemented in the PLD. This core will be used to test the prototype units and determine the maximum performance parameters for the parallel optical interconnect link in the future. Both loop back (self test) and point-to-point links will be tested. Potential users of this high bandwidth interconnect technology, including Sandia's super computing and advanced high speed networking communities, have been contacted and have expressed interest in fielding the results of this research in a test bed. A development plan for future deployment of this technology has been created.

1.4 Accomplishments

In the first year we exceeded all of the goals and met the project milestones including: 1) Operational concepts and target technologies have been developed that will realize a massively parallel optical back plane. At the hardware level, commercially available fiber optic connectors have been identified that can contain as many as 160 optical fibers. The fibers are made up of 12 fiber ribbon cables. A new generation of VCSELs and receivers are being designed and initial devices will be fabricated within Sandia's Compound Semiconductor Research Laboratory. A concept that will be tried is an integrated tx/rx pair that can either be used over separate fibers or can possibly operate bi-directionally over a single fiber. 2) Three optical interconnect architectures have been developed. Utilizing a star interconnect architecture, the most promising of the three is a Computer Interconnect with Distributed Reservation and Scheduling. Simulations of this protocol will be performed in the second year and the protocol will be implemented on the prototype hardware. 3) ATM and datagram operational models have been developed. We are now targeting Utopia 6 (our definition) of 160 Gbps as the standard data rate. This work is years ahead of the ATM Forum Standards Committee. 4) Prototype hardware on a PCI bus card is currently being designed and will be completed before the end of this year. Prototype hardware will be fabricated next year that will consist of a PCI based card with two optical connections. Multiple cards will be plugged into a host system and data will be sent over the optical bus to exercise the architectural concepts. The cards will be interconnected with the optical fibers.

In the second year, due to funding limitations and technology delays, we have met two of the three goals set forth at the beginning of the year. This year substantial progress was made in the development of the integrated VCSEL and detector. The development of a fully integrated receiver/transmitter awaits the completion of a planarization process development activity. It was decided early in the year to progress to a test bed implementation using commercially available VCSEL array components. The result was a PCI card containing a twelve channel VCSEL based interconnect receiver and transmitter modules. The board contains a single programmable logic device (PLD) that interconnects the electro-optical modules and the PCI control bus. Several components proposed for use on the PCI card were discontinued before reaching production status therefore several architectures were examined and at least 3 separate boards designed were undertaken. Sandia's membership in the Optical Internetworking Forum (OIF) gave us access to information on upcoming 10 Gigabit Ethernet (GBE) standards using parallel optical channels. In addition, introduction of true LVDS compatible devices from Altera enabled the design of board that used a PLD to directly drive the optics modules. These two developments, while technically enabling, lead to a delayed design of the PCI card. Therefore, we were not able to complete the performance evaluation of the collision-free architectures.

2. Optical Bus Physical Layer

This bus would consist of two connectors on each printed circuit board that will be connected with other boards in the system in an optical loop fashion. Each connector has 12 optical fibers. The bus can have several possible architectures, some or all of which could exist at the same time. As many as 160 fibers can be allocated as needed for each system. The options are the double counter rotating ring, point-to-point connections and point-to-multi-point connections (broadcast). It is intended that the bus is Reconfigurable on the fly as needed to meet current system demands. Logically, the bus interconnects can be arranged as needed by the system. The configuration is reconfigured by the bus controller, which accepts requests from individual elements within the system. The controller and the slave devices can be the same physical device. The device can be made to boot with the appropriate mission by using a single input line that forces the device to be a master bus controller when tied high (VCC) and slave device when tied low (GND). The master bus controller boots with the right hand interface as outputs and left hand interface as receive with no pass through connections. The slave devices boot up with bus lines passing data from left to right. The exceptions are the lines specifically allocated for clock and boot operation.

(This page intentionally left blank)

3. ATM Bus Protocol

Several competing protocols have been identified as candidates for operation of the back plane. As this juncture, much of the work has centered on Asynchronous Transfer Mode (ATM). Sandia is a member of the ATM Forum, the standards body that develops specifications for ATM. The standards are called UTOPIA Level 1, UTOPIA Level 2, etc. The latest approved specification is UTOPIA Level 4 which is a 10 Gbps 32 bit bus protocols which uses ATM cells consisting of 53 bytes or packets of unspecified length. Currently, the Physical Working Group (PHY WG) of the ATM Forum is working on specifications for UTOPIA Level 5, OC-768, 40 Gbps. Initial contributions have been received for UL5. A Sandian (the primary author of this paper) was chosen to be the editor of the UL5 specification. At each of these jumps in performance, the bus width and the frequency of operation have doubled. At each point the problems associated with noise, signal integrity and cross talk have become worse. This work is targeted at over 160 Gbps, which would be call UTOPIA Level 6 is ATM Forum parlance. There have been suggestions that the next data rate of 100 Gbps might be the next target rate. Only time will tell.

(This page intentionally left blank)

4. Physical Layer Parallel Optical Devices

As part of this project, the development of integrated, parallel optical interconnects has been started. It has been determined that as many as 424 parallel optical links may be needed to achieve the 160 Gbps operation. This data width represents a complete ATM cell and would require operation of the bus at 377 MHz. Development of a proof of concept device will concentrate on a 12 element VCSEL array and a 12-element receiver array. Mixed VCSEL and receiver arrays (6 rx, 6 tx) are also under development. These components, fabricated by the end of year one, will be packaged into commercially available fiber optic connectors for use in a prototype back plane in year two.

Sandia will leverage several unique technologies to achieve low power integrated rx/tx electro-optic devices. : Semi-insulating GaAs substrates are required for reduced circuit parasitics and to provide for flexibility on which (if any) lead is common in the optoelectronic chip. Specifically, the baseline approach allows the anode or cathode of each laser to be tied together on chip or left fully independent to suit the specific CMOS chip or MUX/DMUX driver chip. Oxidized VCSELs developed at Sandia have been shown to have high wall-plug efficiency, high slope efficiency, high quantum efficiency, low threshold current, excellent performance uniformity and yield, symmetric non-astigmatic near-field profile, wide bandwidth.

Metal-semiconductor-metal back-to-back Schottky photodiodes are typically limited in bandwidth by transit time (electron-hole-pair transport to electrode fingers) and by junction capacitance (finger geometry). Specifically, MSM provide large area detection at high speeds in contrast to pin photodiodes which are limited typically by the RC characteristics which depends on the device area. Fabrication of MSMs is somewhat easier and requires fewer process steps (giving higher yield) than pin photodiodes. We note, however, that we are also considering a nipin structure in which the upper nip layers are used for the photodiode and the lower pin layers are used for the VCSEL. Such a design is attractive for its ability to be integrated into the process flow; however, a small detector area will be required for high speed. This either increases the size due to the requisite lenses between the fiber and detector, decreases the bandwidth for lower loss, or increases the cost because of the greater complexity. The authors believe it best to use large area detectors.

MSM PDs can be formed under the VCSEL or even in the semi-insulating GaAs on which the VCSEL is grown. The disadvantage in forming the detector under the VCSEL is that this requires fine-line (~1micron) lithography over significant topography (8~10microns for the etched VCSEL), a difficult and unnecessary problem that will compromise array yield. The disadvantage in using the underlying semi-insulating GaAs is that such devices have very leaky terminal characteristics (degrading extinction ratio and error rate performance) and poor responsivity (degrading link loss).

VCSEL-MSM Schematic Cross-Section

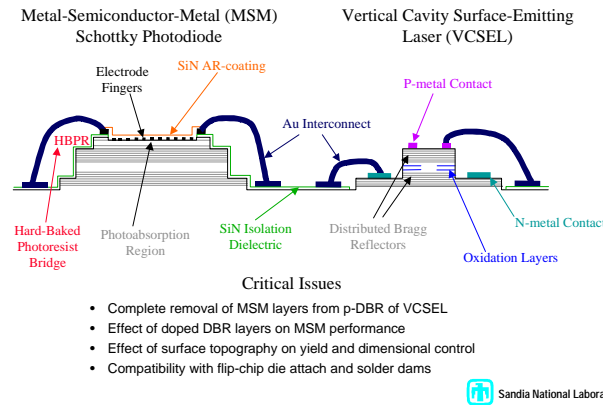


Figure 1. Integrated VCSEL and receiver devices.

The difficulties in the development of the integrated VCSEL/detector are numerous. One problem is the planarization of the surface following fabrication of the multi layer detector stack. MBE growth leaves the surface very rough, with surface artifacts on the order of 12 micrometers. Thin, 1 micrometer wide fingers must be applied to the surface to fabricate the detector with a finger-to-finger pitch of 4 micrometers. This width/pitch is required to achieve the high speed multi gigahertz performance required of this device. The process of fabricating these fingers must be optimized both to insure planarization and to insure that the material used in the fingers will survive the high temperatures experienced in fabrication of the VCSEL stack. Substantial progress has been made in the planarization process. Experiments in the use of refractory metals for the fingers are continuing.

5. Optical Bus / Interconnect System Architecture

In this section, we consider several broadcast-and-select architectures, which can combine the parallel optical technique developed in this LDRD with fiber's unique capability of transmitting light signals of independent wavelengths simultaneously to deliver concurrent transmissions between pairs of interconnected nodes at 200 Gbps each. The optical components needed in these architectures are passive optical couplers, and either a tunable transmitter (TT), a tunable receiver (TR), or both at each interconnected unit. There are three proposed star network architectures with their applications below:

1. Interconnect for Parallel Processors - Tunable Transmitters and Fixed Receivers
2. Computer Interconnect with Multicast capability – A Star Network Using Tunable Receivers
3. Computer Interconnect with Distributed Reservation and Scheduling – WDM Star network with tunable receivers

Although simple, the token-based architecture suffers a severe performance limitation because the token must be processed sequentially by all input and output ports in the network and the architecture is not collision free. Therefore, we propose yet another approach that uses the same configuration as the token architecture (figure 2), which allows nodes on the network to learn all reservations requests and to compute an identical transfer schedules for them in parallel, thereby avoiding collision during the subsequent data transmission. This architecture organizes the data transmission in frame that consists of three phases: reservation, schedule computation, and data transfer (see Figure 3). During the reservation phase, all nodes with data to send will broadcast their request in their time slot on the control network. At the end of the reservation phase, all nodes learn all outstanding data transfer requests for the next frame. Each request includes information such as source address, destination address, and packet length (not needed for ATM applications).

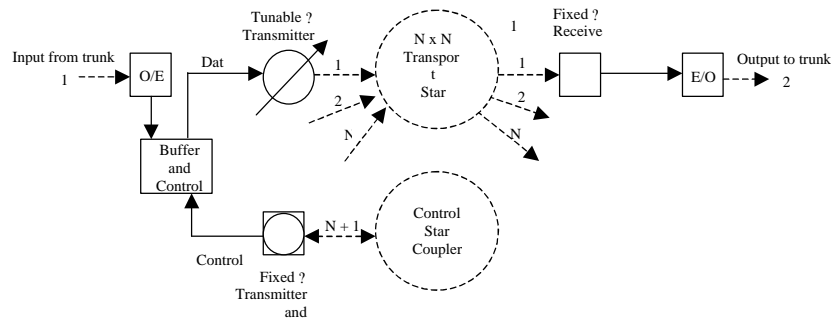


Figure 2. Wavelength addressable cluster interconnect using fast tunable laser and filter

Reservation Phase	Schedule Computation Phase	Data Transfer Phase
-------------------	-------------------------------	---------------------

Figure 3. Structure of a frame

Using a load balanced scheduling algorithm, all nodes will then compute and obtain identical transfer schedules in a collision-free manner. The token processor is shown in Figure 4. At this point, scheduled senders can tune their transmitter to that of the destination and begin the transfer phase using the data network. The duration of this phase is the transmission time of the longest packet in this frame. Note that by overlapping the control phase with the data transfer phase, the network performance in terms of latency and efficiency can be improved significantly. For the remainder of this year, we need to address its implementation issues such as the width of parallel optics for control and data signals, suitable load balanced scheduling algorithms, range of acceptable frame sizes, etc.

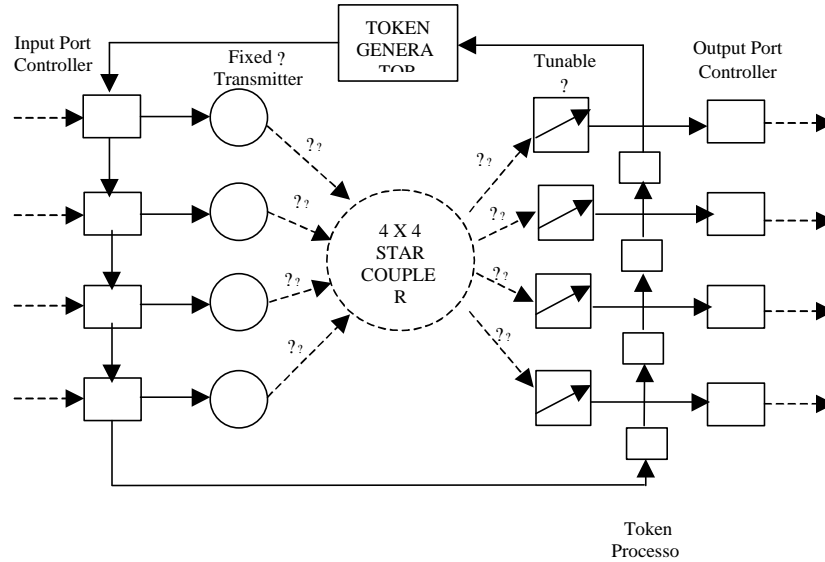


Figure 4. Token Processor

6. Parallel Optical Interconnect PCI Circuit Board

6.1 General

This section describes features of the Parallel Optical Interface (POI) board. The POI board is a multi-channel fiber optic transmitter/receiver and data processor designed to enable development of concepts for 1-10Gbps communication links. Numerous features such as LED displays, auxiliary test connectors, JTAG programmer support, and standalone power inputs have been incorporated to support both development and demonstration activities. This board was developed as part of this project in order to prototype and demonstrate methods, devices and techniques that were invented as part of this project. A line drawing of the POI board is shown in Figure 5. The PCI connector appears at the lower left portion of the card. The parallel optical elements (receiver and transmitter) mount at the left side of the card, the side at the rear of a computer, allowing access to these ports. These elements can be seen clearly in Figure 5 which is the outline drawing of the POI board as seen from the front side. The line drawing of the rear side is shown in Figure 6. The block diagram of the POI board appears in Figure 7.

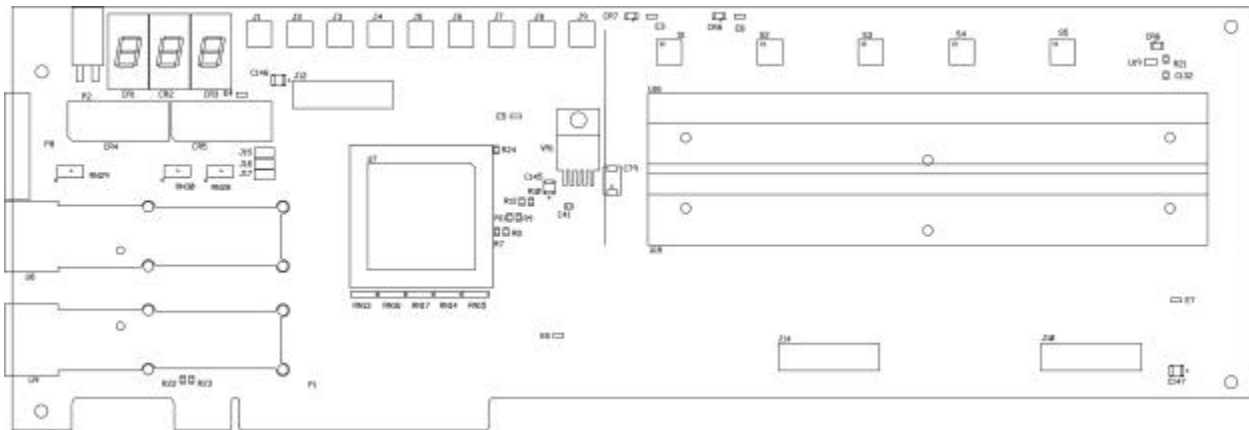


Figure 5. Outline drawing of the POI board, front view.

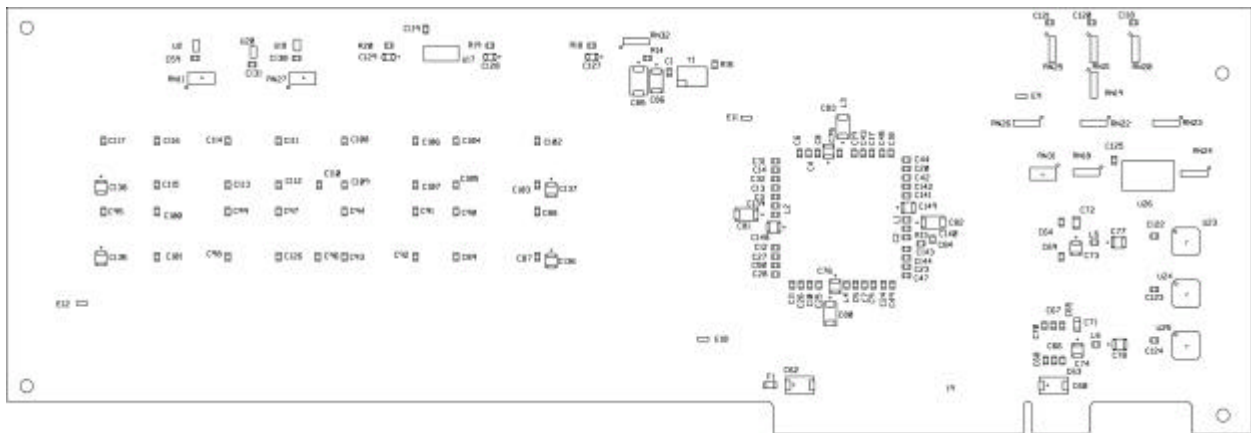


Figure 6. Outline drawing of the POI board, rear view.

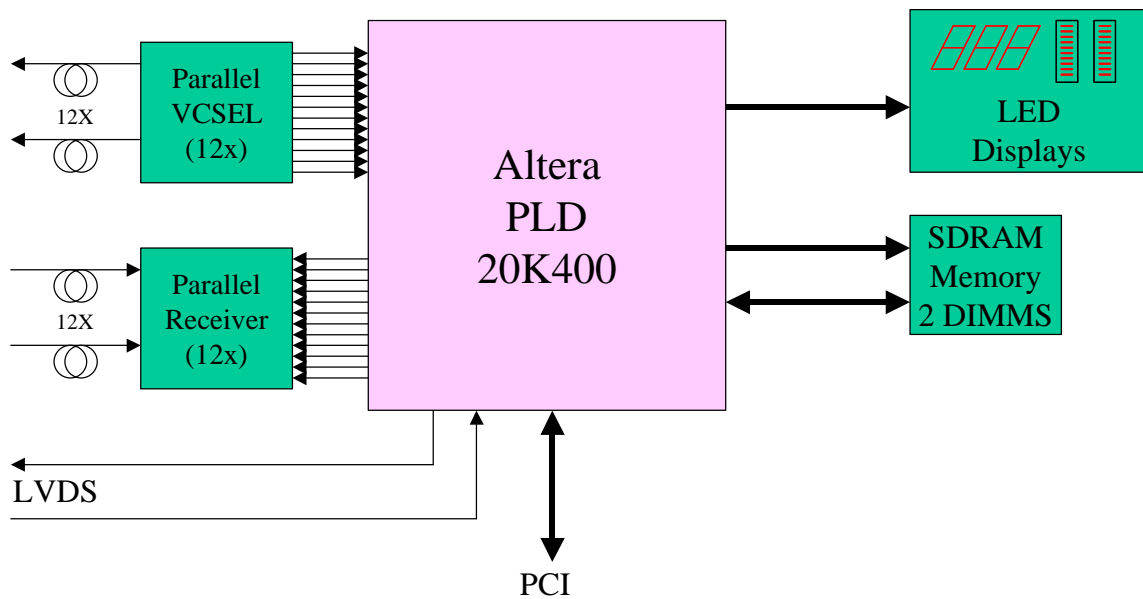


Figure 7. Block diagram of the POI board.

6.2 Board Features

The board has the following features.

- ?? 12 channel TX/RX organized as 11 data +1 clock
- ?? current aggregate data rate of 6.84Gbps (11 x 622.08 Mbps), higher speeds possible
- ?? up to 512Mbytes of dedicated RAM for “at-speed” data source/capture using common PC-100 or PC-133 desktop SDRAM memory
- ?? 32 bit, 33MHz PCI bus interface
- ?? Reconfigurable Altera APEX20KE FPGA for board data processing and control
- ?? flexible test displays including three numeric LEDs and 20 bits of bar graph
- ?? standalone operation
- ?? PCI board form factor
- ?? JTAG programmer input for modifying the Altera FPGA and updating the on-board FPGA programmer
- ?? bundled FPGA programs for:
 - PCI operation
 - basic data channel transmit, receive
 - bit-error rate testing, test pattern source and received data analysis
 - SDRAM control
 - test display operation

The following sections describe the POI board capabilities in more detail. Appendix A lists the bill of materials, Appendix B contains the board schematic, and Appendix C describes the FPGA program created in support of the POI board.

6.3 Fiber Optic Interface

The POI board accepts a 12 fiber ribbon cable using the Infineon V23814-K1306-M130 transmitter and V23815-K1306-M130 receiver modules. The 12 fibers are organized as 11 data fibers and one clock fiber. These parts are designed for data rates of up to 1.25Gbps per fiber, or a maximum of 13.75Gbps.

The Infineon devices convert the fiber signals to/from LVDS signals which are connected to an Altera EP20K400EFC672 FPGA. The EP20K400E interface provides 16 LVDS outputs and 16 LVDS input. In the current generation, each I/O is capable of handling 622.08Mbps. The next generation Altera part will extend the per I/O data rate to over 800Mbps for a aggregate data rate of over 8.8Gbps.

The POI receive and transmit channels are capable of independent operation. Each input and output connects to a high speed 8x serializer/deserializer which converts the 622.08Mbps serial data stream into a 77.76M byte/sec parallel data stream for processing in the Altera FPGA. Clock generation for the transmit/receive channels are handled by separate PLLs that can be operated independently or together allowing synchronous or asynchronous timing.

6.4 Board Design and Layout

The circuit board is made up of 10 conductive layers. The circuit board stack up is shown in Figure 8. There are 6 routing layers and 4 power/ground planes. There are two ground planes and two planes for +3.3V. One of the +3.3V planes is split to accommodate +1.8V needed for the core of the Altera PLD. The Altera PLD is a ball grid device with contact pads on a 1 mm grid. The board was routed using Veribest 98 auto-router with a minimum line width of 0.004 inches (4 mils) and a line spacing of 0.005 inches (5 mils). The exception to this was the routing of the Low Voltage Differential Signal (LVDS) lines, the high speed lines to the optical modules. LVDS lines required careful hand placed routing. These lines were routed on the top layer only, over a specially selected Rogers 4003 dielectric that was 0.020 inches thick. This material was chosen to reduce the signal loss and cross talk on the lines and to enable the controlled impedance, 100 ohm lines to be extremely small. This was a requirement in order to get all the lines routed on the PCI form factor printed circuit board. The lines have a line width of 0.005 inches (5 mils) and each pair of differential lines is separated by 0.004 inches (4 mils). A pair of differential signals is separated from adjacent differential line pairs by 0.025 inches (25 mils). In addition, these lines are routed to insure that all lines are the same length (reducing phase problems) and the corners are all mitered to reduce reflections. It was decided that the corners did not need to be completely rounded at the operational frequency (approximately 1.25 GHz square waves). The rising edges were estimated to be approximately 300 ns). The bandwidth needed for these signals was estimated to be approximately 10 GHz. An example of the LVDS lines is shown in Figure 9.

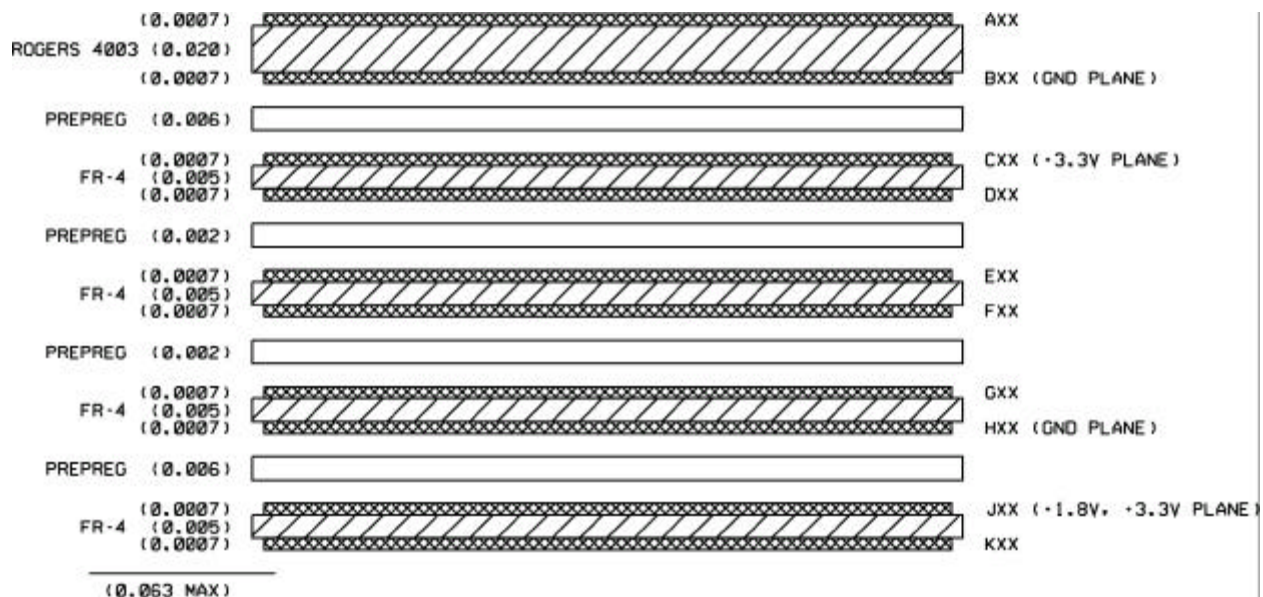
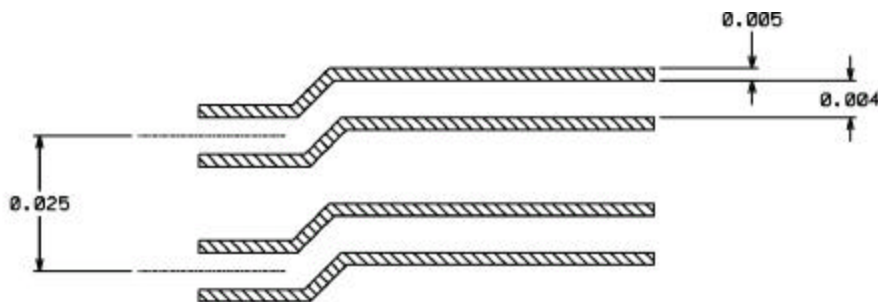


Figure 8. Printed circuit board cross section.



DIFFERENTIAL COUPLED LINES CONTROLLED IMPEDANCE = 100 OHMS

Figure 9. High speed LVDS printed circuit lines.

6.5 Data Processing

Signals to and from the fiber optic interface are converted into 88 bit wide parallel words for use by the Altera FPGA internal logic. The Altera device is capable of 100MHz+ clock rates and will have over 80% of its internal resources available for user defined processing.

Functions included in the standard data processing set include a Sandia National Labs developed bit error rate tester (BERT), a 32 bit, 33 MHz PCI target, an SDRAM controller, and LED display controllers.

The BERT provides three major functions: pseudo-random sequence generation, received data synchronization, and bit error rate estimation. Versions of this function are currently in use in a 0.15 Gb/s OC-3 ATM/SONET Cell Generator/Checker, and a 2.5 Gb/s OC-48c ATM/SONET Cell Generator/Checker. An important feature of the BERT is its ease of scaling to match differing data rates and sequence lengths. Documentation for the BERT includes a technique for choosing generator polynomials that produce the fastest data stream generation.

A PCI target is integrated into the standard FPGA code to support development efforts using a standard desktop computer. An SDRAM controller is used to allow up to 512Mbytes of deep data capture or specialized pattern generation. The controller manages a 128 bit wide data path to standard PC-100 or PC-133, 168 pin DIMM, desktop computer memory and can transfer data at the normal signal processing rate.

6.6 Miscellaneous Features

Altera FPGA LVDS I/Os not connected to the fiber optic modules are arranged as 3 debug data paths, an auxiliary clock and data input, and a clock and data output. The debug paths are wrapped from the Altera outputs back to its inputs. The auxiliary clock and data signals are routed to SMA connectors at the POI board top edge. A separate SMA connector allows access to the on-board crystal oscillator circuit for monitoring or (with the removal of a resistor) replacing the onboard reference clock.

A JTAG connector is provided for user configuration of the Altera FPGA and programming of the Flash ROM. In support of debug and demonstration, the POI board provides three momentary action pushbutton switches, three 7-segment LED numeric displays, and 20 bits of LED bar graph. These functions are all driven from the FPGA allowing full user definition of their representation. Additional switches are provided to generate a board reset and to force the FPGA to reload its program. A discrete LED provides visual feedback on the FPGA program state.

Three standard Mictor test connectors are placed to support debug. Two of the connectors are placed in the FPGA to SDRAM circuit for monitoring memory transactions. The third connector is connected directly to the FPGA for monitoring internal FPGA processes.

6.7 Mechanical & Power

The POI board is implemented in a standard 3.3V, long PCI form factor to allow its use in a desktop computer environment. The POI board also has a power connector to support board operation without a host computer. The POI board will require 3.3V at 4amps (est.)

7. Future Development Activities

7.1 System Demonstration

The parallel optical interconnect technology is ready for a system demonstration. There are several areas that still need attention. In FY00, we have evaluated several vendors of these transmitters and receivers, as well as Sandia's own unique VCSEL (Vertical Cavity Surface Emitting Laser) technology and continue to refine our efforts toward an integrated VCSEL/detector array. The current implementation operates with 12 parallel optical channels at up to 1.2 Gbps. The PLD LVDS lines are limited, however, to 622 Mbps due to lack of availability of higher speed parts from Altera. It is expected that Altera will have 1.25 Gbps components some time in the next year. The optics are capable of operating at 1.25 Gbps. There are also versions of the optic modules that will operate at over 2.5 Gbps. Availability of these components is also limited, but we should see some production of components in the upcoming year.

In the future, we propose to configure our 12 channel optical bus card into a 12 port Gigabit Ethernet PCI card by adding a "simple" routing engine in programmable logic to each card and use them as a distributed switch/compute node for cluster computing. In this scheme, a compute node would be attached to each switch via its PCI buss. Source routing would be used and the routing engine on each card could be as simple as a lookup table. There are many routing schemes possible which would drive the complexity of the ASIC based routing engine on the card. The beauty of this interconnect is that it avoids the central point of failure inherent in a single switch architecture. The path diversity inherent in this scheme lends itself to a fault tolerant system as many alternate paths are available between any set of compute nodes. If done correctly, this same path diversity also would ensure maximum system throughput as blocking could be minimized. Since the PCI bus can only support one or possibly two (depending on flavor) Gig-E interfaces, trying to route traffic from the excess 10 interfaces through the host, or even have the host service the interrupts from all 12 would be impossible. However, a small switch/router included on each card would offload this from the host and make possible a very rich interconnect topology.

7.2 Associated Tasks

- 1) Develop software to implement 12 channel Gigabit Ethernet on our Optical Bus Demonstration Card.
- 2) Develop software to include switch/router function in demo card.
- 3) Iterate optical bus card design.
- 4) Refine processing steps for monolithic VCSEL and detector arrays.
- 5) Demonstrate optical bus in a Cplant-like environment.

(This page intentionally left blank)

8. Summary

The result of this LDRD has been the development of parallel optical devices and module technology. Techniques for implementation of a parallel optical bus have been discussed. Work on an integrated VCSEL and receiver have been discussed. Based on Sandia's unique VCSEL (Vertical Cavity Surface Emitting Laser) technology, these devices are compatible with CMOS (Complementary Metal Oxide Semiconductor) chips and have single channel bandwidth in excess of 20 GHz. This interconnect scheme has been developed as the physical layer of a greater ATM (Asynchronous Transfer Mode) based backplane. There are several advantages to this technology including small board space, lower power and non-contact communication. This technology is also easily expandable to meet future bandwidth requirements in excess of 160 Gbps sometimes referred to as UTOPIA 6. Currently, activities at the ATM Forum and the Optical Internetworking Forum (OIF) have concentrated on the development of standards for OC768 (40 Gbps). One such proposal from the OIF would use 12 channels of VCSEL based lasers operating at 1.25 Gbps each to accomplish a 40 Gbps aggregate channel. In the first year we developed integrated VCSELs and receiver techniques, identified fiber ribbon based interconnect scheme and a high level architecture. In the second year, we implemented the physical layer in the form of a PCI computer peripheral card. A description of future work including super computer networking deployment and protocol processing is included.

(This page intentionally left blank)

9. References

- [1] A quick search of SAND reports over the past 10 years related to VCSELs returns 79 papers.
- [2] SAND94-1558A, “Arrays of red VCSELs with partial top dielectric stack DBRs,” Richard P. Schneider, et al.
- [3] IEEE Std 1596.3 – 1996, “IEEE Standard for Low-Voltage Differential Signals (LVDS) for Scalable Coherent Interface (SCI)”, approved March 21, 1996.
- [4] ANSI/TIA/EIA-644-1995, “Electrical Characteristics of Low Voltage Differential Signaling (LVDS) Interface Circuits”, approved November 15, 1995.
- [5] ATM Forum, www.atmforum.com. OIF, www.oif.com.
- [6] “Startup marries optics, ASICs to attack 10 Gbits/s,” Craig Matsumoto, September 18, 2000, *Electronic Engineering Times*, pg. 42.
- [7] www.stratoslightwave.com
- [8] “Newport brings LVDS to 2.5-Gbit/s realm,” Craig Matsumoto, September 18, 2000, *Electronic Engineering Times*, pg. 48.
- [9] “As InfiniBand nears completion, Lucent, Intel ready support silicon,” Loring Wirbel, September 4, 2000, *Electronic Engineering Times*, pg. 50.
- [10] “Optical Links Eliminate Bulky Box-to-Box Cables,” Dave Bursky, October 30, 2000, *Electronic Design*, pgs. 78-80.
- [11] “Emerging Technologies Series #4: Going Long-Haul with 1300-nm VCSELs,” Andrew Jackson, November, 2000, *Communication Systems Design*, pgs. 77-81.

(This page intentionally left blank)

Appendicies

(This page intentionally left blank)

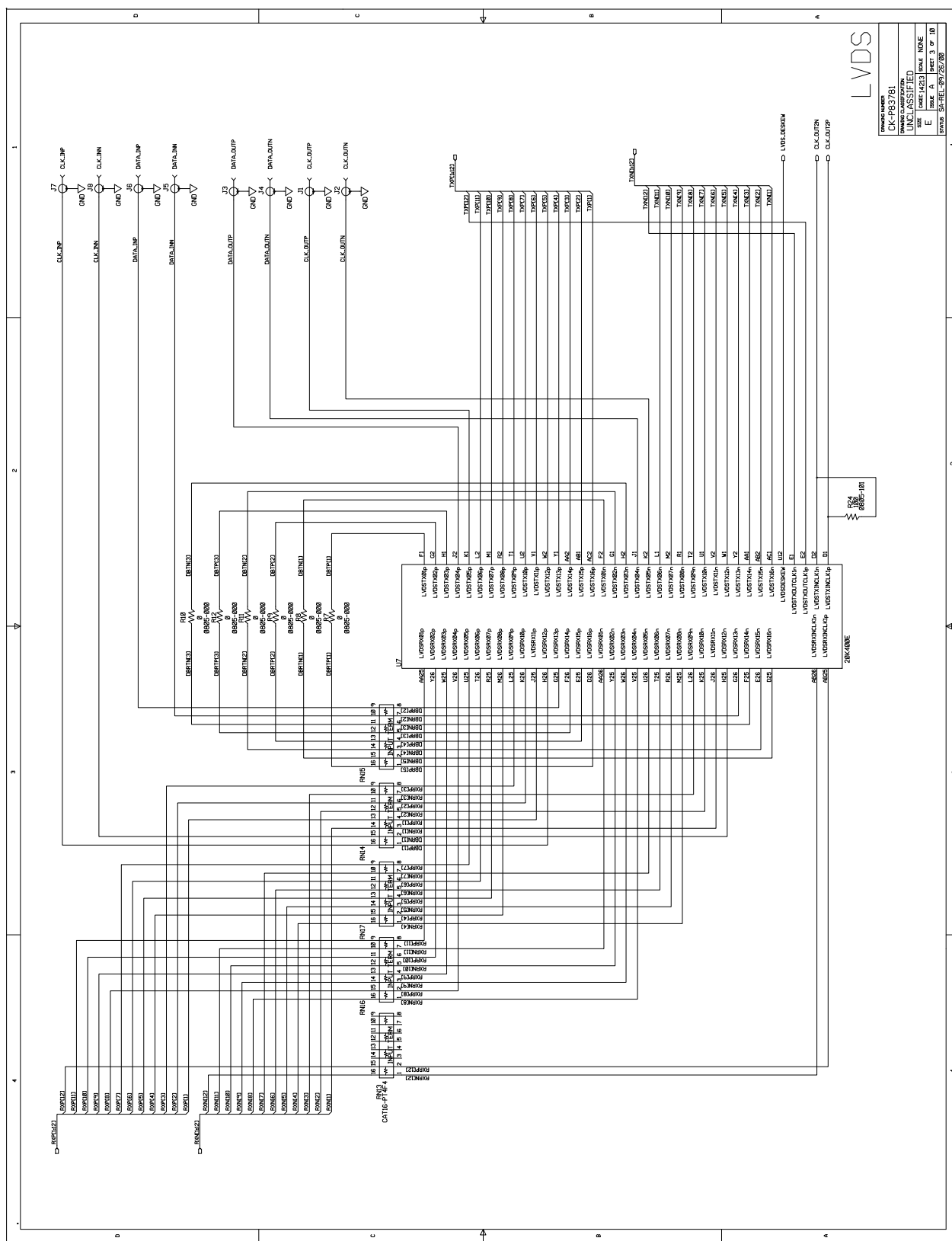
Appendix A. Bill of Materials for POI Board

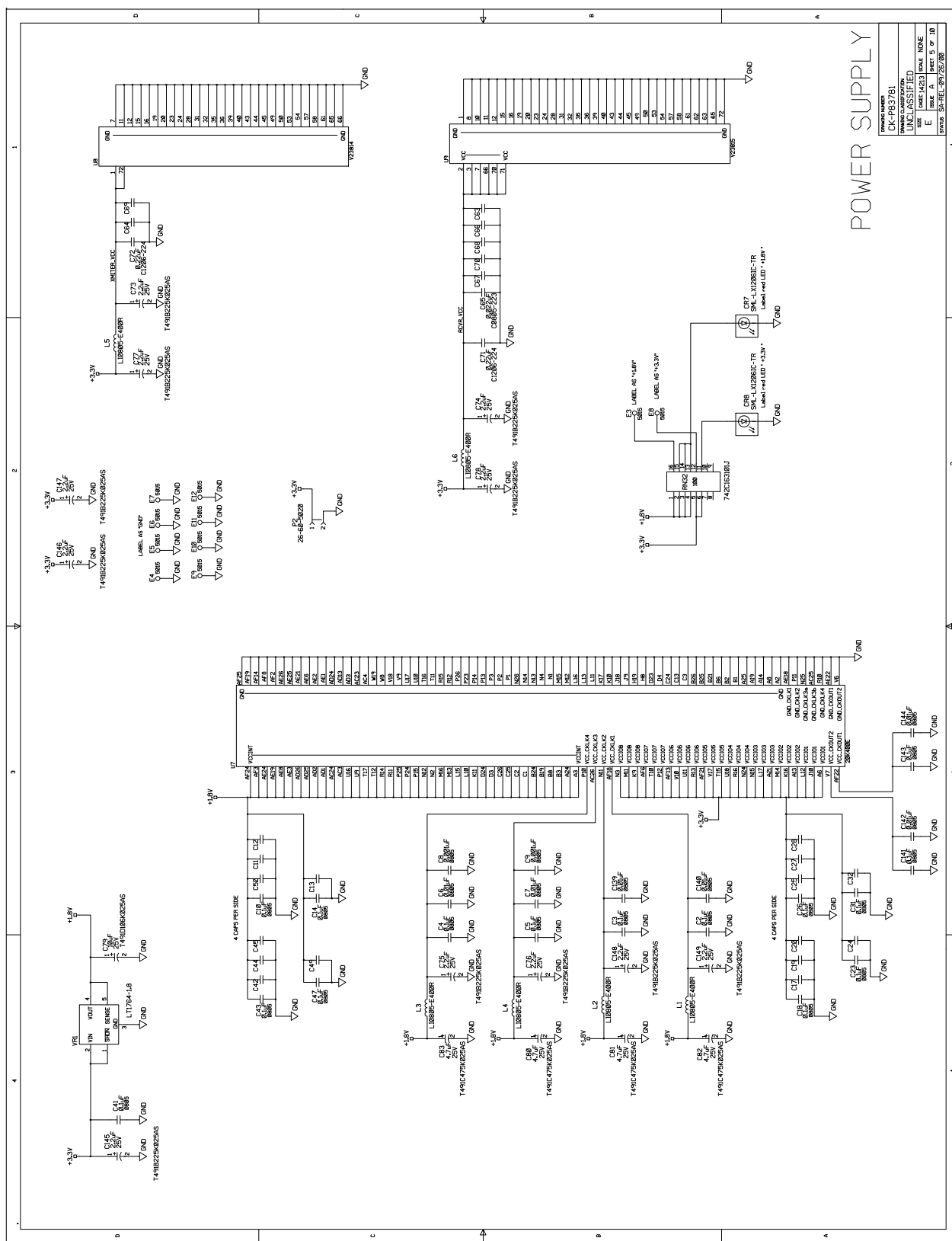
QTY	PART NUMBER	DESCRIPTION	REF DES
5	CAT16-PT4F4	RES. NETWORK ARRAY (BOURNS) RN15,RN16 RN17	RN13,RN14
1	CB3LV-3C-19.4400	OSCILLATOR, CLK, 19.440Mhz (CTS REEVES)	Y1
84	C0805-104	CAP., CE, 0.1uF, 0805	C1-C14,C17 C18-C20,C23 C24-C28,C31 C32,C41,C42 C43-C45,C47 C49,C50,C59 C87-C100 C101,C102 C103,C104 C105,C106 C107,C108 C109,C110 C111,C112 C113,C114 C115,C116 C117,C118 C119,C120 C121,C122 C123,C124 C125,C126 C130,C131 C132,C139 C140,C141 C142,C143 C144
1	C0805-220	CAP., CE, 22pF, 0805	C84
8	C0805-223	CAP., CE, 0.022uF, 0805	C63-C70
2	C1206-224	CAP., CE, 0.22uF, 1206	C71,C72
1	EDGCON-120	EDGE CONNECTOR, 120 PIN	P1
3	EPC2TC32	EEPROM (ALTERA)	U23-U25
5	KT11P2SM	SWITCH, PUSH (C&K)	S1-S5
3	LDS-A304RI	DISPLAY, LED, 7 SEGMENT (LUMEX)	CR1-CR3
6	LI0805-E400R	INDUCTOR, 0805 (STEWART)	L1-L6
1	LT1764-1.8	REGULATOR, LDO (LINEAR TECH.)	VR1
2	MAX6306UK30D3-T	IC, PROGRAMMABLE RESET (MAXIM)	U2,U18
1	MC74LVX14D	HEX SCHMITT INVERTER (MOTOROLA)	U17
2	MT4LSDT	SDRAM MODULE (MICRON)	U15,U16
2	NC7SZD384P5	LED DRIVER (FAIRCHILD)	U19,U20
3	SML-LX1206IC-TR	LED, SMT, RED (LUMEX)	CR6-CR8
1	SN74HC244DW	OCTAL BUFFER/LINE DRIVER (T.I.)	U26
2	SSA-LXB101W-GF/L	LED, 10 SEGMENT (LUMEX INC.)	CR4,CR5
3	T491A334K025AS	CAP., TANT, 0.33uF, 10%, 25V (KEMET)	C127,C128 C129
15	T491B225K025AS	CAP., TANT, 2.2uF, 10%, 25V (KEMET)	C73-C78,C135 C136,C137 C138,C145 C146,C147 C148,C149

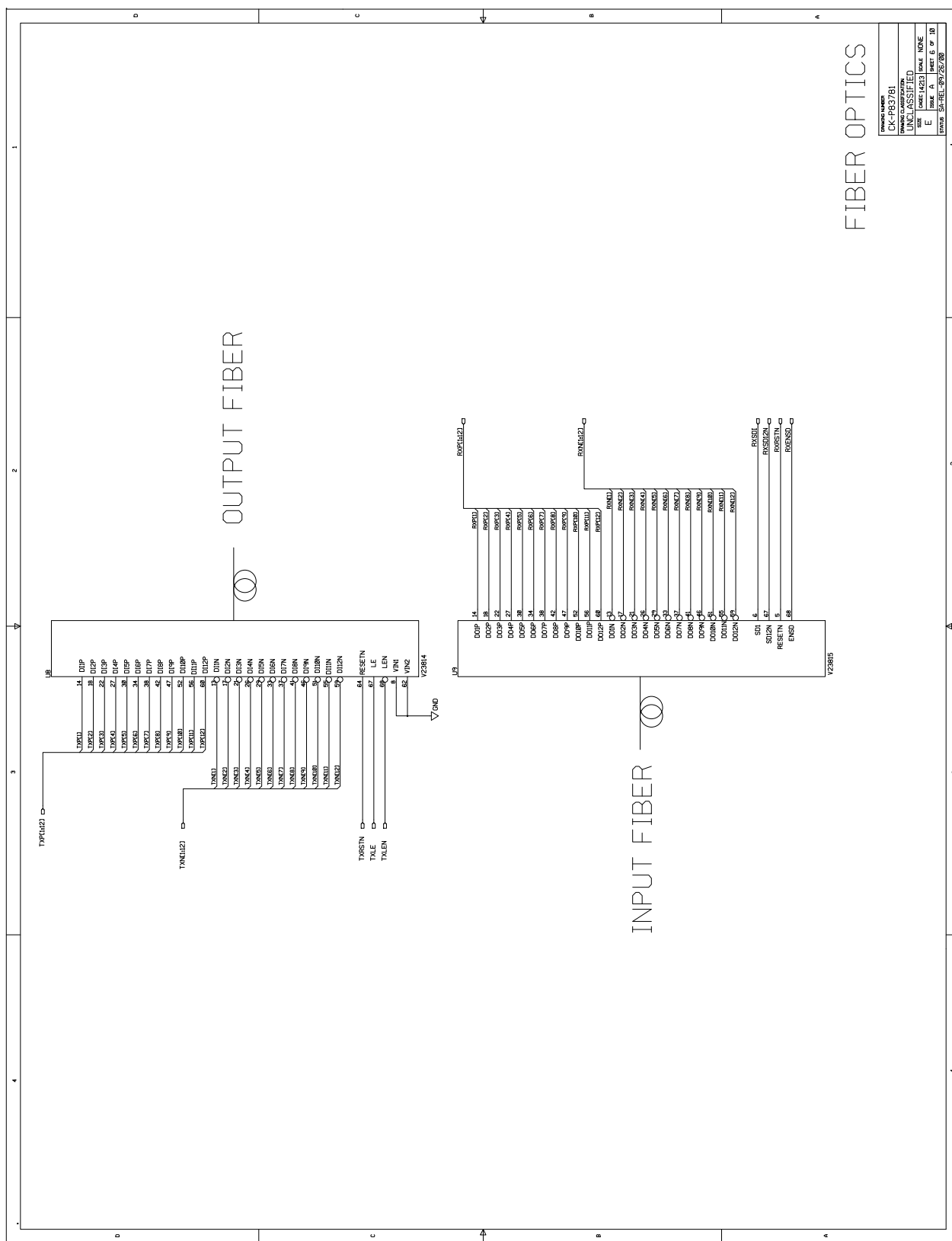
5	T491C475K025AS	CAP., TANT, 4.7uF, 10%, 25V (KEMET)	C80-C83,C86
3	T491D106K025AS	CAP., TANT, 10uF, 10%, 25V (KEMET)	C60,C62,C79
1	T491X107K016AS	CAP., TANT, 100uF, 10%, 16V (KEMET)	C85
1	V23814	PARALLEL OPTICAL TRANSMITTER (SIEMENS)	U8
1	V23815	PARALLEL OPTICAL RECEIVER (SIEMENS)	U9
3	1-87215-0	HEADER, DOUBLE ROW, 2 PINS (AMP)	J15-J17
1	20K400E	PLD, 672 PIN BGA (ALTERA)	U7
1	26-60-5020	HEADER, RT ANGLE, 2 PIN (MOLEX)	P2
1	429.005	FUSE, SMT, 5A (LITTELFUSE)	F1
10	742C163101J	RES. NETWORK, ISOLATED, 100 OHM (CTS)	RN18,RN19 RN20,RN21 RN22,RN23 RN24,RN25 RN26,RN32 RN30,RN31 RN11,RN27 RN28
2	745C101102J	RES. NETWORK, BUSSED, 1K (CTS)	RN29
3	745C101103J	RES. NETWORK, BUSSED, 10K (CTS)	R14
1	745C101222J	RES. NETWORK, BUSSED, 2.2K (CTS)	R7-R12,R22 R23
1	0805-1R0	RES., SMT, 1.0, 0805	R24
8	0805-000	RES., SMT, 0 OHM, 0805	R18-R20 R21
1	0805-101	RES., SMT, 100, 0805	R15
3	0805-104	RES., SMT, 100K, 0805	R13
1	0805-111	RES., SMT, 110, 0805	E3-E12
1	0805-330	RES., SMT, 30 OHM, 0805	P8
1	0805-510	RES., SMT, 51, 0805	J1-J9
10	5015	TESTPOINT, SMT (KEYSTONE)	J10,J12,J14
1	87333-2420	HEADER, RT ANGLE, 24 PIN (MOLEX)	
9	221789-1	CONNECTOR, COAX, STRAIGHT, SMA (AMP)	
3	767054-1	CONNECTOR, MICTOR, 38 PINS (AMP)	

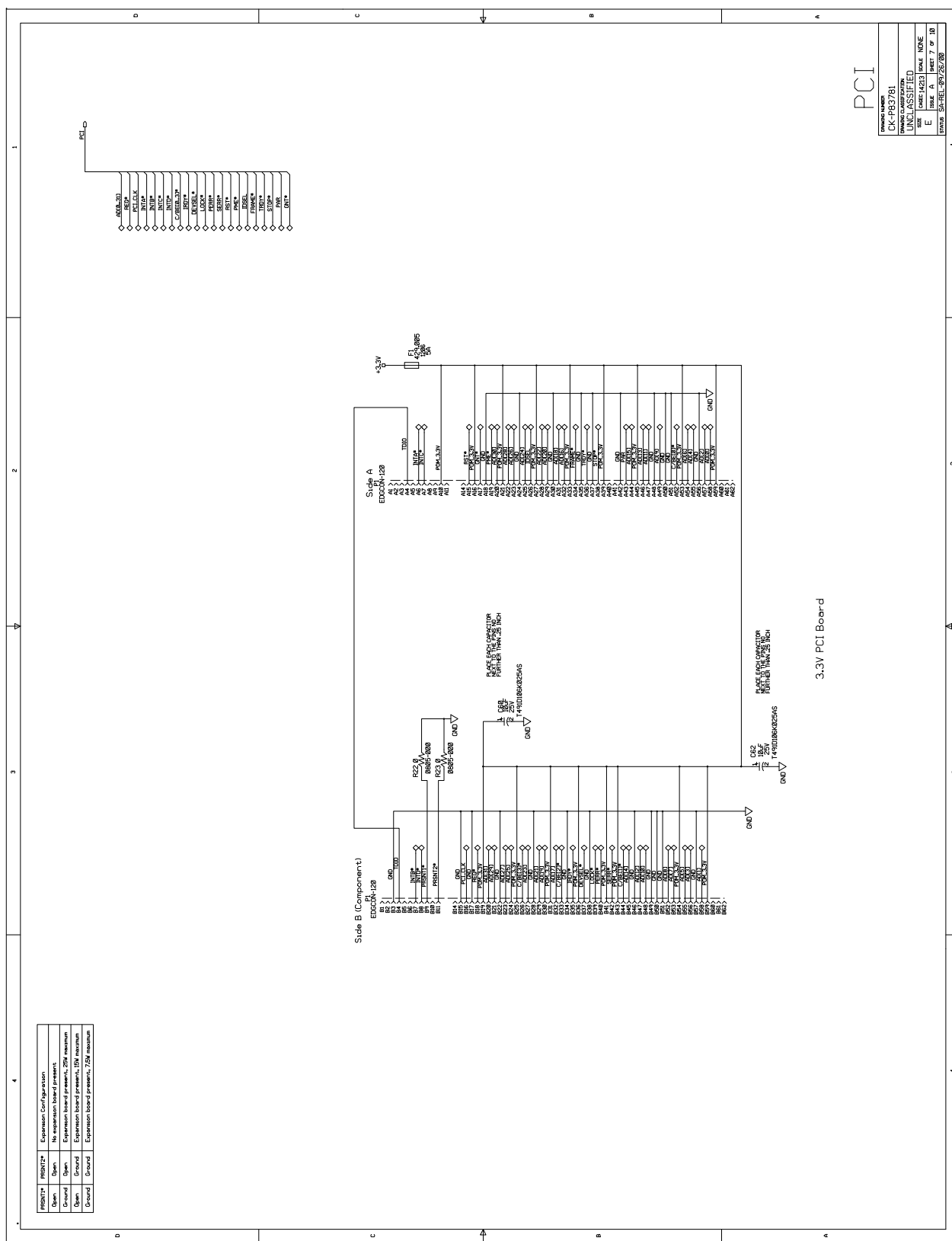
Appendix B. POI Schematic

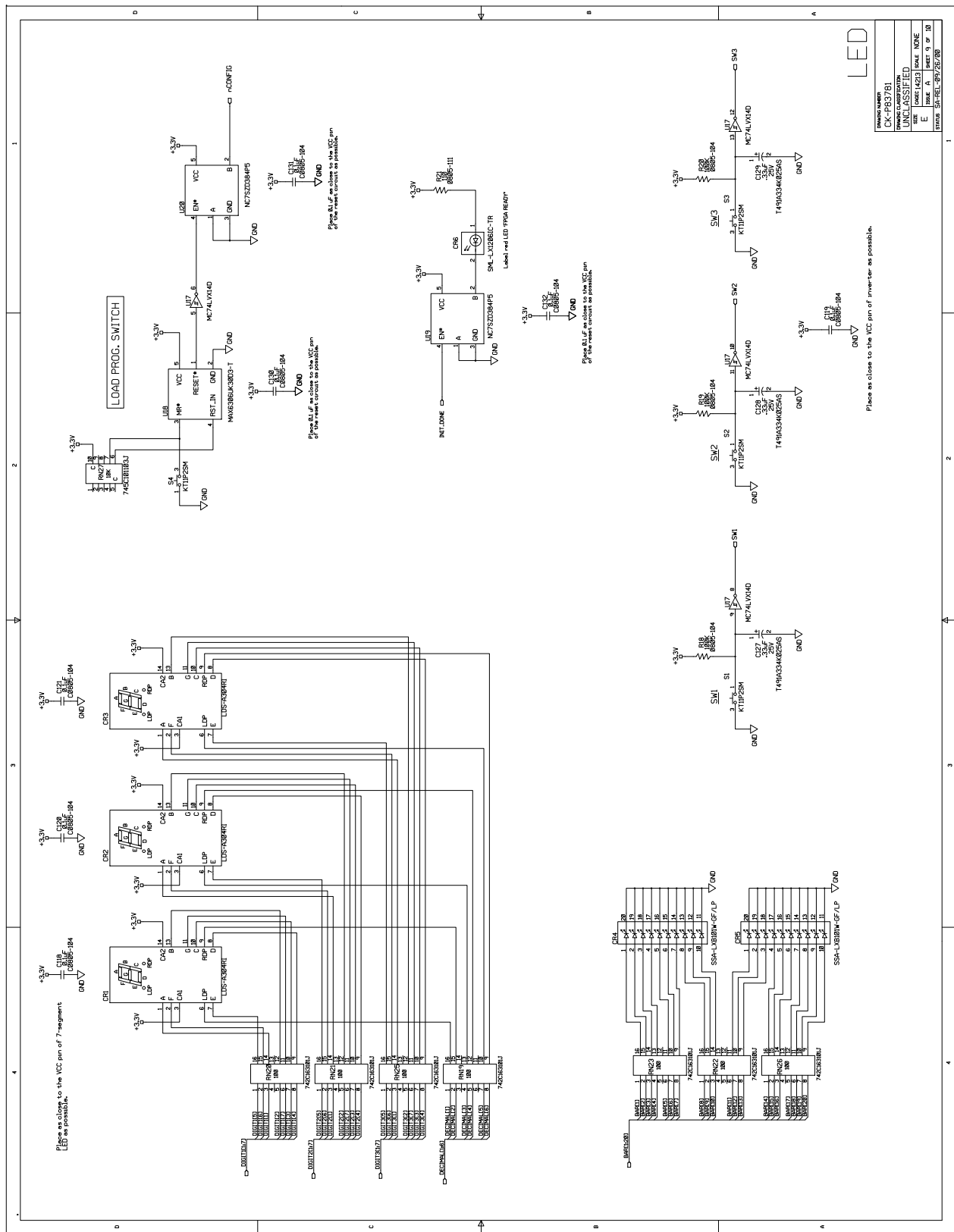
TABLE OF CONTENTS	
PAGE 2	PLD1
PAGE 3	LVDS
PAGE 4	CLOCK
PAGE 5	POWER SUPPLY
PAGE 6	FIBER OPTICS
PAGE 7	PCI
PAGE 8	JTAG ROM PROGRAM
PAGE 9	LED
PAGE 10	SDRAM
MODIFICATIONS:	





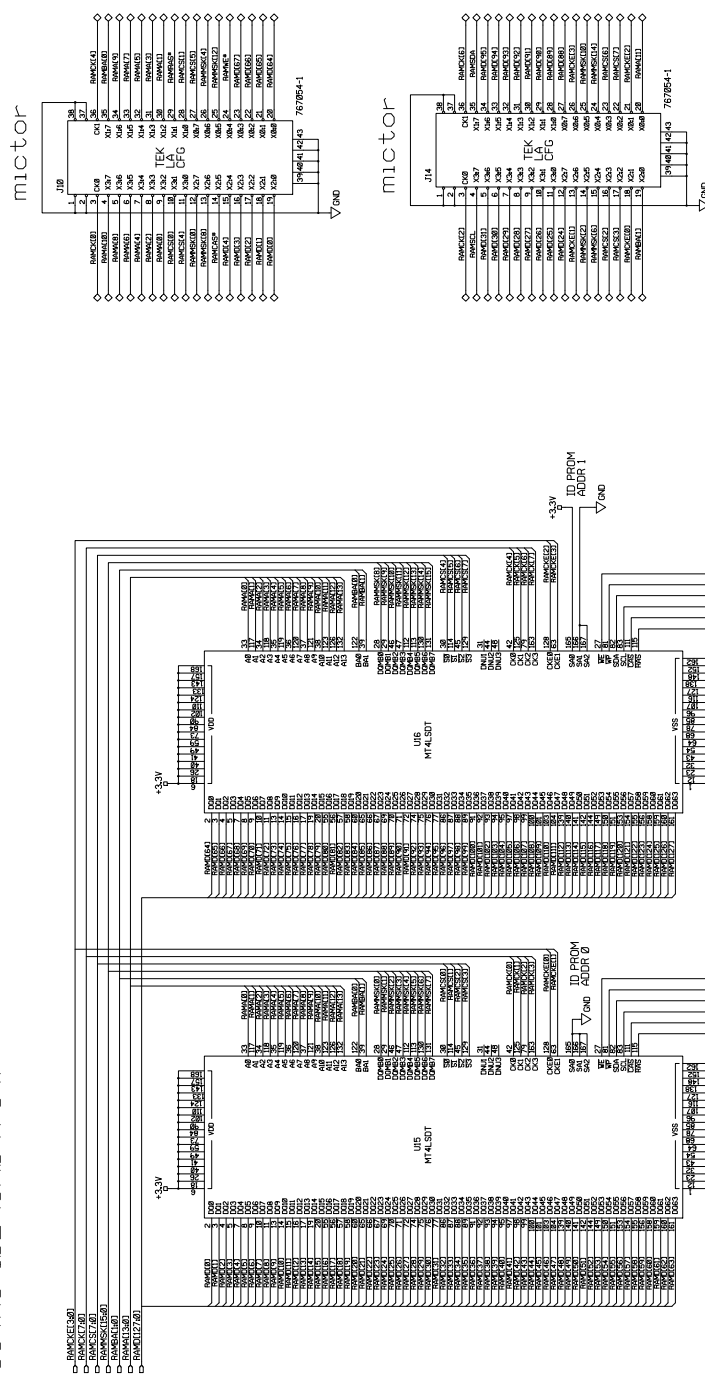






PC100 COMPLIANT, 16 Mb X 64 SDRAM DIMMS
ORGANIZED AS A 16Mb X 124 MEMORY

MICRON PART NUMBERS
MT16LSDT1664AG-10E (16Mb X 64)
MT8LSDT864AG-10E (8Mb X 64)



5

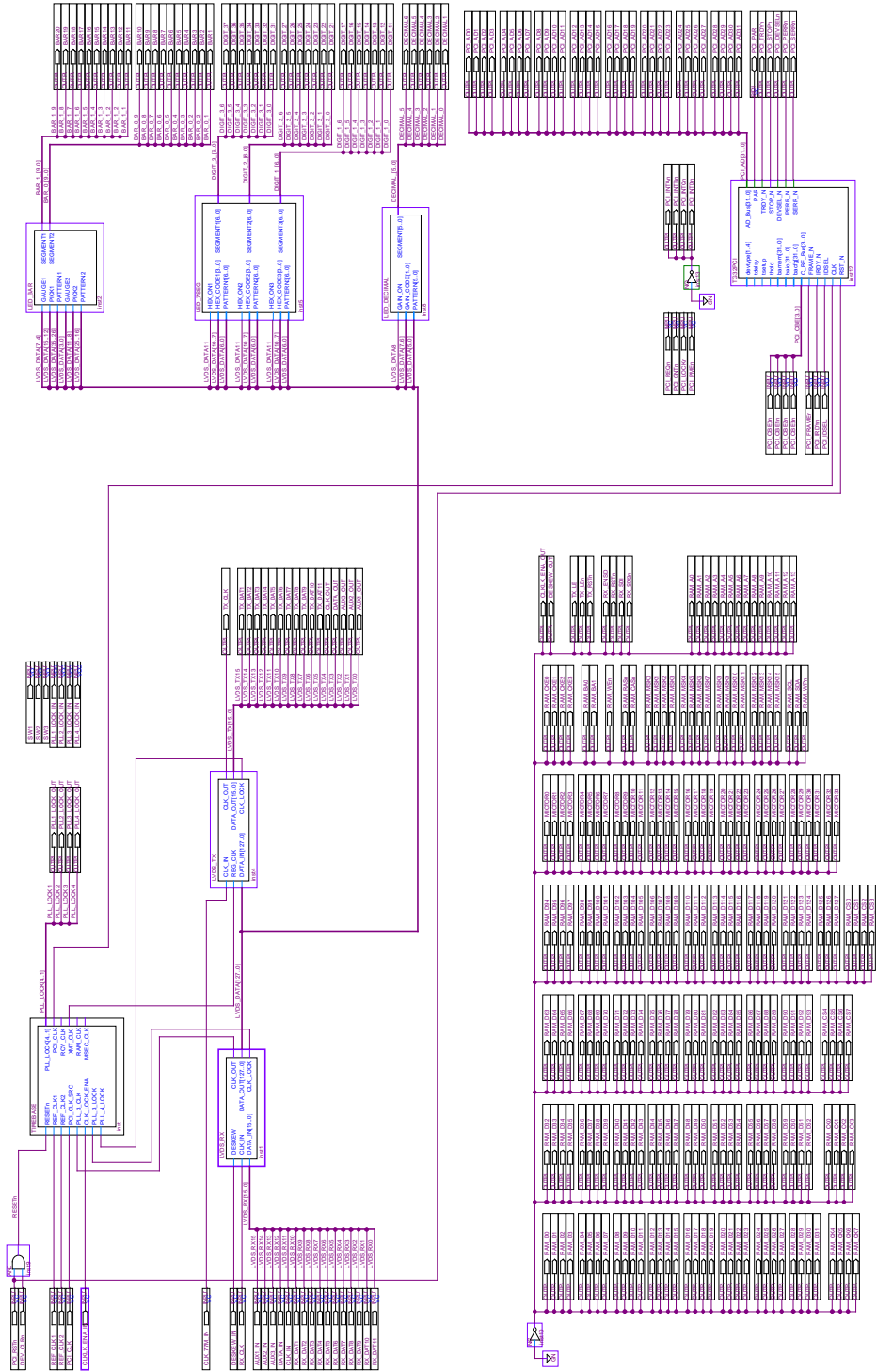
DRAWING NUMBER		CK-P83781	
DRAWING CLASSIFICATION			
UNCLASSIFIED			
SIZE	CAGEC 14213	SCALE	NONE
E	ISSUE A	SHEET	10 OF 10
STATUS SA-REL -09/26/00			

Appendix C. FPGA Program

- C.1 Top Level Block Diagram
- C.2 Common VHDL
- C.3 Timebase VHDL
- C.4 LVDS_RX VHDL
- C.5 LVDS_TX VHDL
- C.6 LED_Bar VHDL
- C.7 LED_7Seg VHDL
- C.8 LED_Decimal VHDL
- C.9 PCI Interface
- C.10 SDRAM Controller

(This page intentionally left blank)

C.1 Top Level Block Diagram



(This page intentionally left blank)

C.2 Common VHDL

The common.vhd package contains interface definitions and common usage definitions for the other VHDL files used in the project. By placing these definitions in a single file, synchronization problems common in the development process are minimized.

```
-----
-- file: common.vhd
-- used: common definitions, those used in multiple modules
-----
-- notes:
-----

library ieee;
use      ieee.std_logic_1164.all;

package COMMON is

    -----
    -- clock definitions
    -----
    --
    -- base system clock
    --
    constant REF_CLK_FREQ      : integer := 19440; --kHz
    constant REF_CLK_PERIOD    : integer := 51440; --ps
    --
    -- LVDS tx/rx clock
    --
    constant LVDS_CLK_FREQ     : integer := 77760; --kHz
    constant LVDS_CLK_PERIOD   : integer := 12860; --ps

    -----
    -- LVDS port definitions
    -----
    constant LVDS_CHANNEL_WIDTH : integer := 16;
    subtype  LVDS_CHANNEL_TYPE   is std_logic_vector(LVDS_CHANNEL_WIDTH-1 downto 0);
    --
    constant LVDS_DESERIALIZE    : integer := 8;
    --
    constant LVDS_DATA_WIDTH     : integer := LVDS_CHANNEL_WIDTH * LVDS_DESERIALIZE;
    subtype  LVDS_DATA_TYPE      is std_logic_vector(LVDS_DATA_WIDTH-1 downto 0);

    -----
    -- LED display definitions
    -----
    --
    -- bar graph
    --
    constant LED_BAR_WIDTH      : integer := 10;
    constant LED_BAR_CODE_WIDTH : integer := 4;
    subtype  LED_BAR_PATTERN_TYPE is std_logic_vector(LED_BAR_WIDTH-1 downto 0);
    subtype  LED_BAR_CODE_TYPE   is std_logic_vector(LED_BAR_CODE_WIDTH-1 downto
0);
    --
    -- 7 segment numeric display

```

```

--
constant LED_7SEG_WIDTH      : integer := 7;
constant LED_7SEG_CODE_WIDTH : integer := 4;
subtype LED_7SEG_PATTERN_TYPE is std_logic_vector(LED_7SEG_WIDTH-1 downto 0);
subtype LED_7SEG_CODE_TYPE   is std_logic_vector(LED_7SEG_CODE_WIDTH-1 downto
0);
--
-- 7 segment decimal display
--
constant LED_DECIMAL_WIDTH      : integer := 6;
constant LED_DECIMAL_CODE_WIDTH : integer := 2;
subtype LED_DECIMAL_PATTERN_TYPE is std_logic_vector(LED_DECIMAL_WIDTH-1 downto
0);
subtype LED_DECIMAL_CODE_TYPE   is std_logic_vector(LED_DECIMAL_CODE_WIDTH-1
downto 0);

-----
-- PCI interface definitions
-----
constant PCI_BUS_WIDTH : integer := 32;
constant PCI_CBE_WIDTH : integer := 4;
subtype PCI_BUS_TYPE   is std_logic_vector(PCI_BUS_WIDTH-1 downto 0);
subtype PCI_CBE_TYPE   is std_logic_vector(PCI_CBE_WIDTH-1 downto 0);

end COMMON;

```

C.3 Timebase VHDL

The Timebase code module implements a single point for the creation and distribution of the various clock and timing signals used in the project. Four dedicated clocks are generated: a 97MHz SDRAM clock, a 77MHz LVDS processing clock, and two 622MHz LVDS clocks (Tx and Rx). In addition to these clocks, a millisecond timebase is generated for use in slower processes.

Due to the unique constraints imposed by the Altera LVDS PLL usage, this module explicitly configures only the APEX 20K400E PLLs 1 and 2. PLLs 3 and 4 are configured by the “LVDS_Rx.vhd” and “LVDS_Tx.vhd” code modules, respectively. The resulting clocks and status are then routed through this module.

```
-----
-- file: timebase.vhd
-- used: instantiates and sets up the APEX 20KE PLLs, generates other
--       timing signals for internal logic
-----
-- notes:
--   this module explicitly configures PLLs 1 & 2. PLLs 3 & 4 are
--   implicitly configured by the altlvds_rx/tx LPMs, (instantiated
--   elsewhere.) PLLs 1 & 2 expect the same frequency.
--   assuming a reference clock input of 19.44MHz:
--       PLL1 - 97.20 MHz internal logic clock, SDRAM clock
--       PLL2 - 77.76 MHz internal LVDS logic clock
--       PLL3 - 622.08 MHz LVDS receive clock (see altlvds_rx LPM)
--       PLL4 - 622.08 MHz LVDS transmit clock (see altlvds_tx LPM)
-----

library ieee;
use      ieee.std_logic_1164.all;

library lpm;
use      lpm.lpm_components.all;

entity TIMEBASE is
  port (
    -- system interface
    RESETn      : in  std_logic;
    REF_CLK1     : in  std_logic;  --ref clk to PLL1
    REF_CLK2     : in  std_logic;  --ref clk to PLL2, same as _SRC1
    PCI_CLK_SRC  : in  std_logic;
    PLL_3_CLK    : in  std_logic;  --from altlvds_rx LPM
    -- PLL control interface
    CLK_LOCK_ENA : in  std_logic;
    -- PLL status interface
    PLL_3_LOCK   : in  std_logic;
    PLL_4_LOCK   : in  std_logic;
    PLL_LOCK     : out std_logic_vector(4 downto 1);
    -- timebase signals
    PCI_CLK      : out std_logic;
    RCV_CLK      : out std_logic;
    XMT_CLK      : out std_logic;
    RAM_CLK      : out std_logic;
```

```

        MSEC_CLK      : out std_logic);
end TIMEBASE;

architecture STRUCTURAL of TIMEBASE is
--
-- Altera APEX 20KE PLL component
--
component altclklock
generic (
    inclock_period      : natural := 1;
    inclock_settings    : string  := "UNUSED";
    valid_lock_cycles   : natural := 5;
    invalid_lock_cycles : natural := 5;
    valid_lock_multiplier : natural := 5;
    invalid_lock_multiplier : natural := 5;
    operation_mode      : string  := "NORMAL";
    clock0_boost        : natural := 1;
    clock0_divide       : natural := 1;
    clock1_boost        : natural := 1;
    clock1_divide       : natural := 1;
    clock0_settings     : string  := "UNUSED";
    clock1_settings     : string  := "UNUSED";
    outclock_phase_shift : natural := 0;
);
port (
    inclock      : in  std_logic := '0';
    inclocken    : in  std_logic := '0';
    fbin         : in  std_logic := '0';
    clock0       : out std_logic;
    clock1       : out std_logic;
    locked       : out std_logic
);
end component;
--
-- input clock definitions
--
constant REF_FREQ      : integer := 19440;          --kHz
constant REF_PERIOD    : natural := (10**9)/REF_FREQ; --ps
--
constant LVDS_FREQ     : integer := 77760;          --kHz
constant LVDS_PERIOD   : natural := (10**9)/LVDS_FREQ; --ps
--
constant SDRAM_FREQ    : integer := 100000;          --kHz
constant SDRAM_PERIOD  : natural := (10**9)/SDRAM_FREQ; --ps
--
-- PLL definitions
--
constant PLL_1_MULT     : natural := SDRAM_FREQ/REF_FREQ;
constant PLL_1_FREQ     : integer := REF_FREQ * PLL_1_MULT; --kHz
constant PLL_1_PERIOD   : integer := (10**9)/PLL_1_FREQ;   --ps
--
constant PLL_2_MULT     : natural := LVDS_FREQ/REF_FREQ;
constant PLL_2_FREQ     : integer := REF_FREQ * PLL_2_MULT; --kHz
constant PLL_2_PERIOD   : integer := (10**9)/PLL_2_FREQ;   --ps
--
-- aux timebase definitions
--

```

```

constant MSEC_FREQ      : integer := 1000;
constant MSEC_MAX       : integer := (REF_FREQ / MSEC_FREQ) - 1;
SUBTYPE MSEC_TYPE is integer range 0 to MSEC_MAX;
signal    MSEC_COUNT    : MSEC_TYPE;
--
-- interconnects
--
signal    PLL_1_CLK      : std_logic;
signal    PLL_1_LOCK     : std_logic;
--
signal    PLL_2_CLK      : std_logic;
signal    PLL_2_LOCK     : std_logic;
--
signal    REF_CLK_INT    : std_logic;
signal    MSEC_CLK_INT   : std_logic;

begin

-----
-- PLL 1
--   reference input x5 to generate high speed internal logic
--   and SDRAM clock, must use the clock1 output
-----
PLL_1_LPM : altclklock
  generic map (
    inclock_period => REF_PERIOD,
    clock0_boost   => PLL_1_MULT
  )
  port map (
    inclock      => REF_CLK1,
    inclocken    => CLK_LOCK_ENA,
    clock0       => PLL_1_CLK,
    locked       => PLL_1_LOCK
  );

-----
-- PLL 2
--   reference input x4 to generate LVDS base clock, must use
--   the clock1 output
-----
PLL_2_LPM : altclklock
  generic map (
    inclock_period => REF_PERIOD,
    clock0_boost   => 1,
    clock1_boost   => PLL_2_MULT
  )
  port map (
    inclock      => REF_CLK2,
    inclocken    => CLK_LOCK_ENA,
    clock0       => REF_CLK_INT,
    clock1       => PLL_2_CLK,
    locked       => PLL_2_LOCK
  );

-----

```

```

-- PLL 3
-- LVDS receive clock
-----

--
-- see altlvds_rx LPM instantiation
--

-----

-- PLL 4
-- LVDS transmit clock
-----

--
-- see altlvds_tx LPM instantiation
--

-----

-- 1 msec timebase
-----

MSEC_TIMER_GEN: process (RESETn, REF_CLK_INT)
begin
    if (RESETn = '0') then
        MSEC_COUNT    <= 0;
        MSEC_CLK_INT <= '0';
    elsif (rising_edge(REF_CLK_INT)) then
        if (MSEC_COUNT = MSEC_MAX) then
            MSEC_COUNT    <= 0;
            MSEC_CLK_INT <= '1';
        else
            MSEC_COUNT    <= MSEC_COUNT + 1;
            MSEC_CLK_INT <= '0';
        end if;
    end if;
end process MSEC_TIMER_GEN;

-----

-- output assignments
-----

--
-- PLL status
--
PLL_LOCK <= (PLL_4_LOCK, PLL_3_LOCK, PLL_2_LOCK, PLL_1_LOCK);
--
-- timebase signals
PCI_CLK  <= PCI_CLK_SRC;
RCV_CLK  <= PLL_3_CLK;
XMT_CLK  <= PLL_2_CLK;
RAM_CLK  <= PLL_1_CLK;
MSEC_CLK <= MSEC_CLK_INT;

end STRUCTURAL;

```

C.4 LVDS_RX VHDL

This module configures the Altera LVDS Receive interface and associated PLL. As currently implemented, this module will accept all 16 LVDS inputs and create a 128 bit parallel output.

```
-----
-- file: lvds_rx.vhd
-- used: instantiate and setup the APEX 20KE LVDS receive function
--       and PLL
-----
-- notes:
--       this module sets up the APEX 20KE PLL3
-----

library ieee;
use      ieee.std_logic_1164.all;

library lpm;
use      lpm.lpm_components.all;

library work;
use      work.common.all;

entity LVDS_RX is
  port (
    -- control inputs
    DESKEW   : in  std_logic;
    -- LVDS inputs
    CLK_IN   : in  std_logic;
    DATA_IN : in  LVDS_CHANNEL_TYPE;
    -- parallel outputs
    CLK_OUT  : out std_logic;
    DATA_OUT : out LVDS_DATA_TYPE;
    -- status outputs
    CLK_LOCK : out std_logic);
end LVDS_RX;

architecture STRUCTURAL of LVDS_RX is
  --
  -- Altera LVDS receive LPM
  --
  component altlvds_rx
  generic (
    number_of_channels      : natural := LVDS_CHANNEL_WIDTH;
    deserialization_factor : natural := LVDS_DESERIALIZE;
    inclock_period          : natural := LVDS_CLK_PERIOD;
    registered_output       : string  := "ON");
  port (
    rx_deskew   : in  std_logic := '0';
    rx_inclock  : in  std_logic := '0';
    rx_in       : in  LVDS_CHANNEL_TYPE := (others => '0');
    rx_outclock : out std_logic;
    rx_out      : out LVDS_DATA_TYPE;
    rx_locked   : out std_logic);
  end component;
end architecture;
```

```

--
-- interconnects
--
signal CLK_OUT_INT  : std_logic;
signal DATA_OUT_INT : LVDS_DATA_TYPE;
signal CLK_LOCK_INT : std_logic;

begin

-----
-- LVDS receiver & PLL3
-----

LVDS_RX_LPM: component altlvds_rx
port map (
    rx_deskew    => DESKEW,
    rx_inclock   => CLK_IN,
    rx_in        => DATA_IN,
    rx_outclock  => CLK_OUT_INT,
    rx_out       => DATA_OUT_INT,
    rx_locked    => CLK_LOCK_INT);
--
-- output assignments
--
CLK_OUT  <= CLK_OUT_INT;
DATA_OUT <= DATA_OUT_INT;
CLK_LOCK <= CLK_LOCK_INT;

end STRUCTURAL;

```

C.5 LVDS_TX VHDL

This module configures the Altera LVDS Transmit interface and associated PLL. As currently implemented, this module will generate all 16 LVDS outputs from a 128 bit parallel input.

```
-----
-- file: lvds_tx.vhd
-- used: instantiate and setup the APEX 20KE LVDS transmit function
--       and PLL
-----
-- notes:
--   this module sets up the APEX 20KE PLL4
-----

library ieee;
use      ieee.std_logic_1164.all;

library lpm;
use      lpm.lpm_components.all;

library work;
use      work.common.all;

entity LVDS_TX is
  port (
    -- parallel inputs
    CLK_IN   : in  std_logic;                --base clock
    REG_CLK  : in  std_logic;                --register clock
    DATA_IN : in  LVDS_DATA_TYPE;
    -- LVDS outputs
    CLK_OUT  : out std_logic;
    DATA_OUT : out LVDS_CHANNEL_TYPE;
    -- status outputs
    CLK_LOCK : out std_logic);
end LVDS_TX;

architecture STRUCTURAL of LVDS_TX is
  --
  -- Altera LVDS transmit LPM
  --
  component altlvds_tx
    generic (
      number_of_channels      : NATURAL := LVDS_CHANNEL_WIDTH;
      deserialization_factor : NATURAL := LVDS_DESERIALIZE;
      inclock_period          : NATURAL := LVDS_CLK_PERIOD;
      registered_input        : STRING  := "ON";
      multi_clock              : STRING  := "ON");
    port (
      tx_out          : out LVDS_CHANNEL_TYPE;
      tx_outclock     : out std_logic;
      sync_inclock    : in  std_logic := '0';
      tx_in           : in  LVDS_DATA_TYPE;
      tx_locked       : out std_logic;
      tx_inclock      : in  std_logic := '0');
```

```

end component;
--
-- interconnects
--
signal CLK_OUT_INT  : std_logic;
signal DATA_OUT_INT : LVDS_CHANNEL_TYPE;
signal CLK_LOCK_INT : std_logic;

begin

-----
-- LVDS transmitter & PLL4
-----

LVDS_TX_LPM: component altlvds_tx
port map (
    sync_inclock => REG_CLK,
    tx_in        => DATA_IN,
    tx_inclock   => CLK_IN,
    tx_out       => DATA_OUT_INT,
    tx_outclock  => CLK_OUT_INT,
    tx_locked    => CLK_LOCK_INT
);
--
-- output assignments
--
CLK_OUT  <= CLK_OUT_INT;
DATA_OUT <= DATA_OUT_INT;
CLK_LOCK <= CLK_LOCK_INT;

end STRUCTURAL;

```

C.6 LED_Bar VHDL

This module converts an input code to a bar graph pattern. Three different input control types are supported: “gauge”, “pick”, and “pattern”. The “gauge” input causes all LEDs with values less than the input code to be illuminated, ex. for an input code of 3, LEDs 0, 1, 2, and 3 would be illuminated. The “pick” inputs causes the selected LED to be illuminated, ex. for an input code of 3, only LED 3 would be illuminated. The “pattern” input causes an LED to be illuminated for each “1” in the hexadecimal input code, ex. for an input code of “0x81”, LEDs 0 and 7 would be illuminated. These codes may be simultaneously applied.

```
-----
-- file: led_bar.vhd
-- used: drives two 10 segment LED bars based on different input
--       controls
-----
-- notes:
--   three different input controls are supported:
--   gauge   - segments <= input code are lit
--   pick    - segment = input code is lit
--   pattern - off/on bit pattern
--   results of the three codes are OR'd to form the output
-----

library ieee;
use      ieee.std_logic_1164.all;
use      ieee.std_logic_unsigned.all;

library work;
use      work.common.all;

entity LED_BAR is
  port (
    -- display control inputs
    GAUGE1   : in  LED_BAR_CODE_TYPE;      --"fuel gauge" readout
    PICK1    : in  LED_BAR_CODE_TYPE;      --"pick one" readout
    PATTERN1 : in  LED_BAR_PATTERN_TYPE;    --"pattern" readout
    --
    GAUGE2   : in  LED_BAR_CODE_TYPE;      --"fuel gauge" readout
    PICK2    : in  LED_BAR_CODE_TYPE;      --"pick one" readout
    PATTERN2 : in  LED_BAR_PATTERN_TYPE;    --"pattern" readout
    -- segment control outputs
    SEGMENT1 : out LED_BAR_PATTERN_TYPE;    --"pattern" readout
    SEGMENT2 : out LED_BAR_PATTERN_TYPE);   --"pattern" readout
  end LED_BAR;

architecture BEHAVIORAL of LED_BAR is
  --
  -- interconnects
  --
  signal BAR_GAUGE1   : LED_BAR_PATTERN_TYPE;
  signal BAR_PICK1    : LED_BAR_PATTERN_TYPE;
  signal BAR_PATTERN1 : LED_BAR_PATTERN_TYPE;
  signal BAR_GAUGE2   : LED_BAR_PATTERN_TYPE;
```

```

signal BAR_PICK2      : LED_BAR_PATTERN_TYPE;
signal BAR_PATTERN2   : LED_BAR_PATTERN_TYPE;

begin

-----
-- fuel gauge decode
-----
--
FUEL_GAUGE1: process(GAUGE1)
    variable TEMP_CODE : integer range 0 to LED_BAR_WIDTH-1;
begin
    TEMP_CODE := conv_integer(GAUGE1);
    for I in LED_BAR_PATTERN_TYPE'low to LED_BAR_PATTERN_TYPE'high loop
        if (I > TEMP_CODE) then
            BAR_GAUGE1(I) <= '0';
        else
            BAR_GAUGE1(I) <= '1';
        end if;
    end loop;
end process;
--
FUEL_GAUGE2: process(GAUGE2)
    variable TEMP_CODE : integer range 0 to LED_BAR_WIDTH-1;
begin
    TEMP_CODE := conv_integer(GAUGE2);
    for I in LED_BAR_PATTERN_TYPE'low to LED_BAR_PATTERN_TYPE'high loop
        if (I > TEMP_CODE) then
            BAR_GAUGE2(I) <= '0';
        else
            BAR_GAUGE2(I) <= '1';
        end if;
    end loop;
end process;

-----
-- pick one decode
-----
PICK_ONE1: process(PICK1)
    variable TEMP_CODE : integer range 0 to LED_BAR_WIDTH-1;
begin
    TEMP_CODE := conv_integer(PICK1);
    for I in LED_BAR_PATTERN_TYPE'low to LED_BAR_PATTERN_TYPE'high loop
        if (I = TEMP_CODE) then
            BAR_PICK1(I) <= '1';
        else
            BAR_PICK1(I) <= '0';
        end if;
    end loop;
end process;
--
PICK_ONE2: process(PICK2)
    variable TEMP_CODE : integer range 0 to LED_BAR_WIDTH-1;
begin
    TEMP_CODE := conv_integer(PICK2);
    for I in LED_BAR_PATTERN_TYPE'low to LED_BAR_PATTERN_TYPE'high loop

```

```

        if (I = TEMP_CODE) then
            BAR_PICK2(I) <= '1';
        else
            BAR_PICK2(I) <= '0';
        end if;
    end loop;
end process;

-----
-- pattern decode
-----

BAR_PATTERN1 <= PATTERN1;
BAR_PATTERN2 <= PATTERN2;

-----
-- segment control generator
-----

SEGMENT1 <= BAR_GAUGE1 or BAR_PICK1 or BAR_PATTERN1;
SEGMENT2 <= BAR_GAUGE2 or BAR_PICK2 or BAR_PATTERN2;

end BEHAVIORAL;

```

(This page intentionally left blank)

C.7 LED_7Seg VHDL

This module converts an input code to a 7 segment numerical LED pattern. Two different input control types are supported: “hex” and “pattern”. The “hex” input causes the corresponding number to be displayed, ex. for an input code of 3, a “3” would be displayed, (segments A, B, C, D, and G). An additional input “HEX_ON”, allows the hex display to be disabled. The “pattern” input causes an LED to be illuminated for each “1” in the hexadecimal input code, ex. for an input code of “0x41”, segments A and G would be illuminated. These codes may be simultaneously applied. The decimal segments of the display are controlled in a separate code module, “LED_Decimal.vhd”.

```
-----
-- file: led_7seg.vhd
-- used: drives a 7 segment LED digit based on different input
--       controls
-----
-- notes:
--   two different input controls are supported:
--     hex       - 0x00-0x0F code to segments
--     pattern    - off/on bit pattern
--   results of the two codes are NOR'd to form the output
-----

library ieee;
use      ieee.std_logic_1164.all;

library work;
use      work.common.all;

entity LED_7SEG is
  port (
    -- display control inputs
    HEX_ON1   : in  std_logic;           --"digit" on/off
    HEX_CODE1 : in  LED_7SEG_CODE_TYPE;  --"digit" readout
    PATTERN1   : in  LED_7SEG_PATTERN_TYPE; --"pattern" readout
    --
    HEX_ON2   : in  std_logic;           --"digit" on/off
    HEX_CODE2 : in  LED_7SEG_CODE_TYPE;  --"digit" readout
    PATTERN2   : in  LED_7SEG_PATTERN_TYPE; --"pattern" readout
    --
    HEX_ON3   : in  std_logic;           --"digit" on/off
    HEX_CODE3 : in  LED_7SEG_CODE_TYPE;  --"digit" readout
    PATTERN3   : in  LED_7SEG_PATTERN_TYPE; --"pattern" readout
    -- segment control outputs
    SEGMENT1   : out LED_7SEG_PATTERN_TYPE; --pattern output
    SEGMENT2   : out LED_7SEG_PATTERN_TYPE; --pattern output
    SEGMENT3   : out LED_7SEG_PATTERN_TYPE; --pattern output
  )
end LED_7SEG;

architecture DATAFLOW of LED_7SEG is
  --
  -- hex to segment conversion
```

```

--
-----gfedcba
constant HEX_OFF : LED_7SEG_PATTERN_TYPE := "0000000";
constant HEX_0   : LED_7SEG_PATTERN_TYPE := "0111111";
constant HEX_1   : LED_7SEG_PATTERN_TYPE := "0000110";
constant HEX_2   : LED_7SEG_PATTERN_TYPE := "1011011";
constant HEX_3   : LED_7SEG_PATTERN_TYPE := "1001111";
constant HEX_4   : LED_7SEG_PATTERN_TYPE := "1100110";
constant HEX_5   : LED_7SEG_PATTERN_TYPE := "1101101";
constant HEX_6   : LED_7SEG_PATTERN_TYPE := "1111101";
constant HEX_7   : LED_7SEG_PATTERN_TYPE := "0000111";
constant HEX_8   : LED_7SEG_PATTERN_TYPE := "1111111";
constant HEX_9   : LED_7SEG_PATTERN_TYPE := "1100111";
constant HEX_A   : LED_7SEG_PATTERN_TYPE := "1110111";
constant HEX_B   : LED_7SEG_PATTERN_TYPE := "1111100";
constant HEX_C   : LED_7SEG_PATTERN_TYPE := "0111001";
constant HEX_D   : LED_7SEG_PATTERN_TYPE := "1011110";
constant HEX_E   : LED_7SEG_PATTERN_TYPE := "1111001";
constant HEX_F   : LED_7SEG_PATTERN_TYPE := "1110001";
--
-- interconnects
--
signal SEG_HEX1      : LED_7SEG_PATTERN_TYPE;
signal SEG_PATTERN1  : LED_7SEG_PATTERN_TYPE;
signal SEG_HEX2      : LED_7SEG_PATTERN_TYPE;
signal SEG_PATTERN2  : LED_7SEG_PATTERN_TYPE;
signal SEG_HEX3      : LED_7SEG_PATTERN_TYPE;
signal SEG_PATTERN3  : LED_7SEG_PATTERN_TYPE;

```

begin

```

-----
-- hexadecimal decode
-----

```

```

SEG_HEX1 <= HEX_OFF when (HEX_ON1 = '0' ) else
    HEX_0   when (HEX_CODE1 = "0000") else
    HEX_1   when (HEX_CODE1 = "0001") else
    HEX_2   when (HEX_CODE1 = "0010") else
    HEX_3   when (HEX_CODE1 = "0011") else
    HEX_4   when (HEX_CODE1 = "0100") else
    HEX_5   when (HEX_CODE1 = "0101") else
    HEX_6   when (HEX_CODE1 = "0110") else
    HEX_7   when (HEX_CODE1 = "0111") else
    HEX_8   when (HEX_CODE1 = "1000") else
    HEX_9   when (HEX_CODE1 = "1001") else
    HEX_A   when (HEX_CODE1 = "1010") else
    HEX_B   when (HEX_CODE1 = "1011") else
    HEX_C   when (HEX_CODE1 = "1100") else
    HEX_D   when (HEX_CODE1 = "1101") else
    HEX_E   when (HEX_CODE1 = "1110") else
    HEX_F;
SEG_HEX2 <= HEX_OFF when (HEX_ON2 = '0' ) else
    HEX_0   when (HEX_CODE2 = "0000") else
    HEX_1   when (HEX_CODE2 = "0001") else
    HEX_2   when (HEX_CODE2 = "0010") else
    HEX_3   when (HEX_CODE2 = "0011") else
    HEX_4   when (HEX_CODE2 = "0100") else

```

```

        HEX_5    when (HEX_CODE2 = "0101") else
        HEX_6    when (HEX_CODE2 = "0110") else
        HEX_7    when (HEX_CODE2 = "0111") else
        HEX_8    when (HEX_CODE2 = "1000") else
        HEX_9    when (HEX_CODE2 = "1001") else
        HEX_A    when (HEX_CODE2 = "1010") else
        HEX_B    when (HEX_CODE2 = "1011") else
        HEX_C    when (HEX_CODE2 = "1100") else
        HEX_D    when (HEX_CODE2 = "1101") else
        HEX_E    when (HEX_CODE2 = "1110") else
        HEX_F;
SEG_HEX3 <= HEX_OFF when (HEX_ON3    = '0'    ) else
        HEX_0    when (HEX_CODE3 = "0000") else
        HEX_1    when (HEX_CODE3 = "0001") else
        HEX_2    when (HEX_CODE3 = "0010") else
        HEX_3    when (HEX_CODE3 = "0011") else
        HEX_4    when (HEX_CODE3 = "0100") else
        HEX_5    when (HEX_CODE3 = "0101") else
        HEX_6    when (HEX_CODE3 = "0110") else
        HEX_7    when (HEX_CODE3 = "0111") else
        HEX_8    when (HEX_CODE3 = "1000") else
        HEX_9    when (HEX_CODE3 = "1001") else
        HEX_A    when (HEX_CODE3 = "1010") else
        HEX_B    when (HEX_CODE3 = "1011") else
        HEX_C    when (HEX_CODE3 = "1100") else
        HEX_D    when (HEX_CODE3 = "1101") else
        HEX_E    when (HEX_CODE3 = "1110") else
        HEX_F;

-----
-- pattern decode
-----

SEG_PATTERN1 <= PATTERN1;
SEG_PATTERN2 <= PATTERN2;
SEG_PATTERN3 <= PATTERN3;

-----
-- segment control generator
-----

SEGMENT1 <= not(SEG_HEX1 or SEG_PATTERN1);
SEGMENT2 <= not(SEG_HEX2 or SEG_PATTERN2);
SEGMENT3 <= not(SEG_HEX3 or SEG_PATTERN3);

end DATAFLOW;

```

(This page intentionally left blank)

C.8 LED Decimal VHDL

This module converts an input code to pattern of decimals on the three POI board 7 segment LEDs. Two different input control types are supported: “gain” and “pattern”. The “gain” input assumes that the code being input to the 7 segment displays is a fraction of 1.000. The “gain” code corresponds to a power of 10, where a code of 0 would cause the leftmost decimal to be illuminated and a code of 3 causes the rightmost decimal to be illuminated. When combined with the 7 segment module “hex” inputs the net effect is a “XXXeY” display. The “pattern” input causes an LED to be illuminated for each “1” in the hexadecimal input code, ex. for an input code of “0x21”, would cause the leftmost and rightmost LEDs to be illuminated. These codes may be simultaneously applied. The numeric segments of the display are controlled in a separate code module, “LED_7Seg.vhd”.

```
-- file: led_decimal.vhd
-- used: drives the decimal point on a bank of three 7 segment LED
-- displays based on different input controls
-----
-- notes:
--     two different input controls are supported:
--         gain      - places decimal based on 10^gain, where gain = 0
--                     implies that the left most decimal is on
--         pattern - off/on bit pattern
--     results of the two codes are NOR'd to form the output
-----

library ieee;
use      ieee.std_logic_1164.all;

library work;
use      work.common.all;

entity LED_DECIMAL is
    port (
        -- display control inputs
        GAIN_ON      : in  std_logic;           --"10x gain" on/off
        GAIN_CODE    : in  LED_DECIMAL_CODE_TYPE; --"10x gain" scaling
        PATTERN      : in  LED_DECIMAL_PATTERN_TYPE; --"pattern" readout
        -- segment control outputs
        SEGMENT      : out LED_DECIMAL_PATTERN_TYPE); --pattern output
    end LED_DECIMAL;

architecture DATAFLOW of LED_DECIMAL is
    --
    -- gain to segment conversion
    --
    --
    --
    --
    --
    --
    constant GAIN_OFF : LED_DECIMAL_PATTERN_TYPE := "000000";
    constant GAIN_1   : LED_DECIMAL_PATTERN_TYPE := "100000";
```

```

constant GAIN_10    : LED_DECIMAL_PATTERN_TYPE := "001000";
constant GAIN_100   : LED_DECIMAL_PATTERN_TYPE := "000010";
constant GAIN_1000  : LED_DECIMAL_PATTERN_TYPE := "000001";
constant GAIN_ERR   : LED_DECIMAL_PATTERN_TYPE := "111111";
--
-- interconnects
--
signal SEG_GAIN      : LED_DECIMAL_PATTERN_TYPE;
signal SEG_PATTERN   : LED_DECIMAL_PATTERN_TYPE;

begin

-----
-- hexadecimal decode
-----
SEG_GAIN <= GAIN_OFF  when (GAIN_ON   = '0' ) else
            GAIN_1    when (GAIN_CODE = "00") else
            GAIN_10   when (GAIN_CODE = "01") else
            GAIN_100  when (GAIN_CODE = "10") else
            GAIN_1000 when (GAIN_CODE = "11") else
            GAIN_ERR;

-----
-- pattern decode
-----
SEG_PATTERN <= PATTERN;

-----
-- segment control generator
-----
SEGMENT <= not(SEG_GAIN or SEG_PATTERN);

end DATAFLOW;

```

C.9 PCI Interface

The project plan intended that the PCI interface be implemented using a public domain PCI target core. The most likely core located during the project was found at the <http://www.opencores.org> website. This core was downloaded and evaluated for use on the POI board. Other commercially available cores were found, ex. at the Altera website, but these cores were not considered for use on this project.

As the core was evaluated it became apparent that the core was intended as a high level simulation tool rather than an implementable code module. The code module accepts the standard PCI interface signals but uses non-implementable timing constructs such as “Stop_N <= ‘0’ after 4ns” and does not provide a “backend” or “local” bus interface for integration of user functionality.

A code listing is not provided here but the entire OpenCores download package is included on the soft copy media that accompanies this document.

C.10 SDRAM Controller

The project plan intended that the SDRAM controller be implemented using either public domain or legacy cores. The most likely core located during the project was again at the <http://www.opencores.org> website. This core was downloaded and evaluated for use on the POI board. As an alternative a ZTEC developed SDRAM controller was also examined. Other commercially available cores were found, ex. at the Altera website, but these cores were not considered for use on this project.

The OpenCores Verilog code module was intended to control discrete SDRAM devices rather than the DIMM modules used on the POI board. Conversion of the module to DIMM compatibility would request extension of modules address decoding capabilities and device control outputs. This work was not performed during the project. An additional feature of this code source is the test bench code provided in the download.

The ZTEC SDRAM controller module was written to interface with DIMM modules such as those used on the POI project. This module was not integrated into FPGA program during the project.

Code listings of the two SDRAM controller modules are not provided here but the entire OpenCores download package and ZTEC module are included on the soft copy media that accompanies this document.

UNLIMITED RELEASE

INITIAL DISTRIBUTION:

8	MS0874	P.J. Robertson, 1751
1	MS0874	D. Palmer, 1751
1	MS1071	M. Knoll, 1730
1	MS1079	D. Williams, 1700
2	MS0603	C.T. Sullivan, 1743
1	MS9011	H.Y. Chen, 8910
1	MS9011	J.M. Brandt, 8910
4	MS0874	K. Gass, 1751
2	MS0806	L.G. Pierson, 9336
1	MS0806	E.L. Witzke, 9336
1	MS0801	M.O. Vahle, 9300
1	MS0806	L. Stans, 9336
1	MS9018	Central Technical Files, 8945-1
2	MS0899	Technical Library, 9616
1	MS0612	Review & Approval Desk, 9612 For DOE/OSTI